NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

PROBABILITY OF DETECTION CALCULATIONS USING MATLAB

by

Wei Yung-Chung

June 1993

Thesis Advisor: Gurnam S. Gill

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1993	

4. TITLE AND SUBTITLE

PROBABILITY OF DETECTION CALCULATIONS USING MATLAB

5. FUNDING NUMBERS

6. AUTHOR(S)

Wei Yung-Chung

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the US Government

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release,
distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

A set of highly efficient computer programs based on the Marcum and Swerling's analysis on radar detection has been written in MATLAB to evaluate the probability of detection. The programs are based on accurate methods unlike the detectability method whch is based on approximation. This thesis also outlines radar detection theory and target models as a background.

The goal of this effort is to provide a set of efficient computer programs for student usage and teacher's aid. Programs are designed to be user friendly and run on personal computers.

14. SUBJECT TERMS	15. NUMBER OF PAGES
	109
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500 Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std 239-18

PROBABILITY OF DETECTION CALCULATION USING MATLAB

by

Wei Yung-Chung
Lieutenant Commander, Chinese Navy
B.S., Chinese Navy Academy, 1984

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1993

Author: ______________________________
Wei Yung-Chung

Approved by: ______________________________
G.S. Gill, Thesis Advisor

____
R. Janaswamy, Second Reader

____
Michael A. Morgan, Chairman
Department of Electrical and Computer Engineering

ABSTRACT

A set of highly efficient computer programs based on the Marcum and Swerling's analysis on radar detection has been written in MATLAB to evaluate the probability of detection. The programs are based on accurate methods unlike the detectability method which is based on approximation. This thesis also outlines radar detection theory and target models as a background.

The goal of this effort is to provide a set of efficient computer programs for student usage and teacher's aid. Programs are designed to be user friendly and run on personal computers.

TABLE OF CONTENTS

I. INTRODUCTION

A. BACKGROUND

The Marcum-Swerling (M-S) models are the most commonly used target models in modern radar analysis. The other target models have also developed over period of time, such as chi-square (Weinstock), log-normal. However in this thesis only M-S models will be discussed.

Histrocially, the earliest descriptions of a target were in terms of a single cross-sectional area value. This quantity was usually some type of an average cross section over the aspect angles which the system designer considered most probable. The approach led the system designers to associate a single cross-sectional area of unvarying value, with a target and thus led to the model of a *steady-state target*. Following this concept, systems were designed to achieve a somewhat arbitrarily specified signal-to-noise ratio at some specified maximum operating range. The inherent basic assumption was that the required detection capability and parameter estimation accuracy could be achieved if this goal was met. The dominant rule of the human observer in early detection systems tended to make this approach acceptable. As the requirements on radar systems became more demanding, and as automatic processing of radar returns was developed, the situation underwent a change. The desire to optimize radar designs established the need for more precise target models.

The initial work of J.I.Marcum of the RAND Corporation in the late 1940s applied the work of Rice of the Bell Telephone Laboratories, relating to steady-state signals immersed in noise, to the problem of radar signal detection. Marcum effectively gave a complete treatment to the statistical problem of a group of constant amplitude signal pulses in the presence of noise. His work resulted in an evaluation of previously untabulated functions and a direct application of the results to a gamut of problems in automatic detection. The statistical approach used in Marcum's studies is the basis for much of the work that followed in the radar detection area.

The detection of signals resulting from a fluctuating target is basically different than for signals resulting from a non-fluctuating target. A requirement remained, therefore, to produce the same type of analysis and supporting tabulation of basic functions for a fluctuating target as had been done by Marcum for the nonfluctuating target. This work was done by Swerling, in the early 1950s. He extended Marcum's approach by employing four target models and two different density functions in conjunction with two extremes of correlation.

Swerling's case 1 and case 2 are the target models which describe large complex targets, such as aircraft, rain clutter, and terrain clutter. They represent two extremes of correlation, and the statistical model is derivable from the physical scattering characteristics of such bodies. It can also be stated that empirically derived data on such targets are in very good agreement with those obtained from the mathematical model.

At the time Swerling was doing his work, it was realized that the model suitable for large complex targets would not give an adequate description of large simple structures. The problem of selecting a model to describe this type or class of target is a difficult one to handle directly. There is no obvious parallel development for it from the scattering characteristics of a specific body type. The approach employed was to establish qualitatively the basic statistical behavior of the target cross sections of interest. Having done this, a convenient form of distribution was selected.

Swerling picked a class of distributions know as the chi-square class, which has the exponential density function distribution as one member. By an appropriate selection of a class parameter, namely, the number of degrees of freedom, the qualitative properties desired for the distribution associated with large simple structures was obtained. Swerling's cases 3 and 4 represent targets that behave as if they have four degrees of freedom and are valid for targets such as rockets, missiles, and space-based satellites.

Swerling's two classes of density functions evaluated at two extremes of correlation, together with Marcum's constant target model (case 5), have been the bases of virtually all radar detection analyses.

Five target models according to Marcum-Swerling scheme are as follows:

a) Swerling case 1 - The echo pulses received from a target on any one scan are of constant amplitude throughout the entire scan but are independent (uncorrelated) from scan to scan. This assumption ignores the effect of the antenna beam shape on the echo amplitude. The probability density function of the radar cross section σ is given by the exponential density function.

$$w(\sigma,\overline{\sigma}) = \frac{1}{\overline{\sigma}} e^{(-\sigma/\overline{\sigma})}, \qquad \sigma > 0 \tag{1}$$

where $\overline{\sigma}$ is the average radar cross section

b) Swerling case 2 - With the same density function as case 1 but the fluctulations are more rapid than in case 1 and are taken to be independent from pulse to pulse rather than from scan to scan.

c) Swerling case 3 - The fluctuation is assumed to be independent from scan to scan, as in case 1, but the probability density function is given by the chi-square distribution with four degrees of freedom.

$$w(\sigma,\overline{\sigma}) = \frac{1}{(K-1)!} \frac{K}{\overline{\sigma}} (K\sigma/\overline{\sigma})^{K-1} e^{-(K\sigma/\overline{\sigma})} = \frac{4\sigma}{\overline{\sigma}^2} e^{-2\sigma/\overline{\sigma}}, \quad (K=2) \tag{2}$$

d) Swerling case 4 - With the same density function but the fluctuation is pulse-to-pulse according to Eq(2).

e) Marcum's model (case 5) - With the constant density function which represents steady-state target.

The probability density function of equation (1), applies to a complex target consisting of many independent scatterers of approximately equal echo areas. The probability density function assumed in case 3 and 4 is more indicative of targets that can be represented as one large reflector together with other small reflectors.

There are curves available that can be used to calculate the probability of detection for each of Swerling cases, but for parameters between those charted, the designer has to interpolate. This can be inaccurate sometimes. Therefore it is convenient and useful to provide accurate programs to calculate for any parameters needed by the user.

B. RADAR RECEIVER MODEL

1. DESCRIPTION

The typical super heterodyne radar receiver and the mathematically receiver model is depicted in Fig 1. The difference between square-law and linear envelope detector is that a square-law envelope detector is used in the small signal optimum receiver and a linear envelope detector is used in the large-signal receiver. For mathematical convenience, the square-law detector is applied, but the performance of both detectors is practically the same for all signal-to-noise ratios.

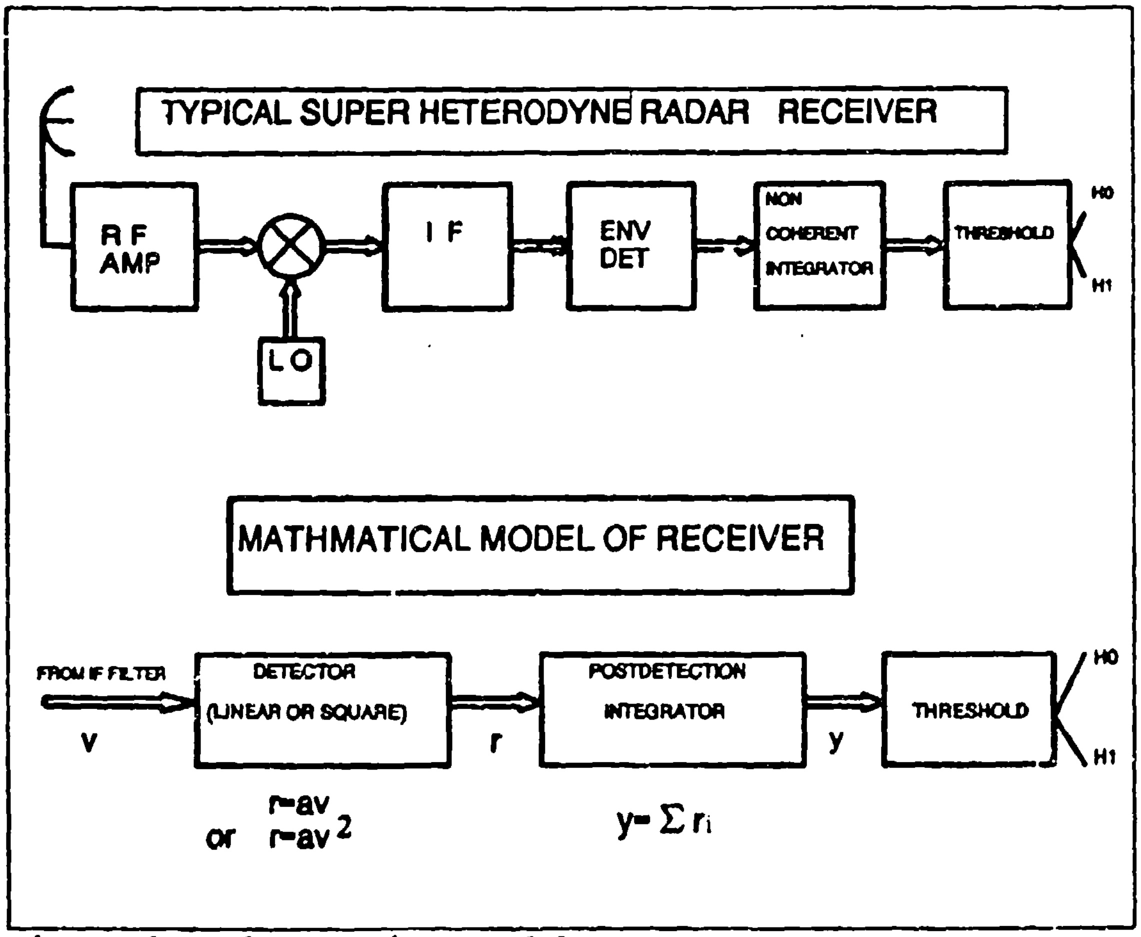

Figure 1: Radar Receiver Model

The multiple-pulse detection is used to improve the detectability of the signal by achieving integration gain. The noncoherent integration provides an integration gain even when the signal has a random phase and is rapidly fluctuating, such as Swerling case 2 and 4. By contrast, coherent integration needs a nonfluctuating or slowly fluctuating signal with predictable phase characteristics.

2. ASSUMPTIONS

The assumptions embodied in the M-S detection problems are as follows:

- A received pulse train consisting of N samples of noise or signal-plus-noise is available.
- The signal is imbedded in white Gaussian noise of known spectral density of $N_0/2$.
- The signal is of unknown phase type, where the RF phase between pulses in the train are randomly distributed.
- The processing consists of a matched filter and square-law envelope detector which operates on each pulse of the train and a linear integration which combines the n square-law envelope detected pulse.
- The directivity pattern of the antenna is rectangular so that the n return pulses resulting during the antenna's dwell time on the target are unaffected by the antenna's radiation pattern.
- The target's cross section can be described by a chi-squared distribution with 2k degree of freedom.

$$w(\sigma,\bar{\sigma}) = \frac{1}{(K-1)!}\frac{K}{\bar{\sigma}}\left(\frac{K\sigma}{\bar{\sigma}}\right)^{K-1} e^{-(K\sigma/\bar{\sigma})} \qquad (3)$$

C. RADAR DETECTION PHILOSOPHY

Radar detection is complicated by the fact that the target cross section σ is a random variable fluctuating with time and both noise and clutter. Extraction, or parameter estimation, is likewise complicated by random fluctuations of the target echo. For the radar detection case Neyman-Pearson criterion is used which needs neither a priori probabilities nor cost estimates. In radar terminology whose objective is to maximize the probability of detection for a given probability of false alarm. This objective can be accomplished by using a likelihood ratio test. Specially, there exists some nonnegative number η such that hypothesis H_1 (i.e.,target is present) is chosen when

$$\Lambda(y) = \frac{f_{s+n}(y)}{f_n(y)} \geq \eta \tag{4}$$

and hypothesis H_0 (no target present) is chosen otherwise. There are two types of errors which can be made. If we say a target is present when in fact it is not, an error of the first kind is made (see Fig 2). That is, we choose H_1 given that H_0 is true which is denoted as the probability of false alarm. Similarly an error of the second kind is made when H_0 is chosen and in fact H_1 is true which is probability of miss.

Declaration \ Actual	H0	H1
H0	Correct decision No target present	Missing target Error of second kind
H1	False alarm Error of first kind	Correct decision Target present

Figure 2: Error of detection

To optimally detect the signal with unknown phase, consider the narrowband signal of duration T given by

$$\begin{aligned} s(t) &= Aa(t)\cos[w_0 t+\theta(t)+\phi] \\ &= s_I\cos\phi - s_Q\sin\phi \end{aligned} \tag{5}$$

where $S_I=Aa(t)\cos[w_0 t+\theta(t)]$, $S_q=Aa(t)\sin[w_0 t+\theta(t)]$, A is signal amplitude, $a(t)$ is an envelope function of duration T, $\theta(t)$ is the signal phase modulation. Waveform $r(t)$ is assumed to be prefiltered by an ideal low-pass filter that is distortionless within $(-w_c, w_c)$ and zero outside the interval; therefore the filter will not affect the band-limited input signal $s(t)$ but results in a band-limited statistical process. Under this assumption the input waveform $r(t)$ can be described by the sampling theorem as

$$r_i(t) = s_i(t) + n_i(t) \qquad i = 1,2,3,\ldots,2f_cT \tag{6}$$

Where $2f_cT$ is the number of samples, and $t=i/2f_c$.

The hypothesis testing can be described as:

$$\begin{aligned} H_1 &: \mathbf{r} = \mathbf{s} + \mathbf{n} \quad (\ target\ present\ ) \\ H_0 &: \mathbf{r} = \mathbf{n} \quad (\ target\ absent\ ) \end{aligned} \tag{7}$$

Then under each hypothesis the probability density functions can be written as:

$$f_{s+n}(\mathbf{r}) = \frac{1}{(\sqrt{2\pi}\sigma_n)^m}\, e^{-\frac{\sum_{i=1}^{m}(r_i-s_i)^2}{2\sigma_n^2}} \; ; \; f_n(\mathbf{r}) = \frac{1}{(\sqrt{2\pi}\sigma_n)^m}\, e^{-\frac{\sum_{i=1}^{m} r_i^2}{2\sigma_n^2}} \tag{8}$$

The likelihood ratio $\Lambda(\mathbf{r})$ is given by

$$\begin{aligned} \Lambda(\mathbf{r}) &= \frac{f_{s+n}(\mathbf{r})}{f_n(\mathbf{r})} = \lim_{\substack{m\to\infty \\ 1/f_c\to 0 \\ m/f_c=T}} \frac{\exp\left(-\sum_{i=1}^{2f_cT} \frac{[r_i-s_i]^2}{2f_cN_0}\right)}{\exp\frac{\left(-\sum_{i=1}^{2f_cT} r_i^2\right)}{2f_cN_0}} \\ &= \frac{\exp\left(-\frac{1}{N_0}\int_0^T [r(t)-s(t)]^2\, dt\right)}{\exp\left(-\frac{1}{N_0}\int_0^T r(t)^2 dt\right)} \\ &= \exp\left[-\frac{1}{N_0}\int_0^T s^2(t)\, dt + \frac{2}{N_0}\int_0^T r(t)s(t)\, dt\right] \end{aligned} \tag{9}$$

where $E = \int_0^T s^2(t)dt$ is the energy of the signal. Substituting Eq(5) into Eq(9) yield the following expression for the likelihood ratio $\Lambda(\mathbf{r}\,|\phi\,)$

$$\Lambda(\boldsymbol{r}|\phi) = \exp\{-\frac{E}{N_0}+\frac{2}{N_0}[\int_0^T r(t)\,s_i(t)\,dt]\cos\phi$$

$$-\frac{2}{N_0}[\int_0^T r(t)\,s_q(t)\,dt]\sin\phi\,\} \tag{10}$$

Let the signal-to-noise ratio $\phi = E/N_0$ and

$$y_I(T) = \frac{2}{N_0\sqrt{\Psi}}\int_0^T r(t)\,s_I(t)\ dt$$

$$y_Q(T) = \frac{2}{N_0\sqrt{\Psi}}\int_0^T r(t)\,s_Q(t)\ dt \tag{11}$$

using Eq(11), Eq(10) can be written as follows

$$\Lambda(\boldsymbol{r}|\phi) = e^{-\Psi/2}\exp\{\sqrt{\Psi}[y_I(T)\ \cos\phi - y_Q(T)\ \sin\phi]\}$$

$$= e^{-\Psi/2}\exp\{\sqrt{\Psi}\ r(T)\cos(\phi+\alpha)\} \tag{12}$$

where $\alpha=\tan^{-1}(y_Q/y_I)$, $r(T)=[y_Q(T)+y_I(T)]^{1/2}$ is the envelope of the radar signal out of a filter matched to the waveform $(2/N_0\ \psi^{1/2})s(t)$, sampled at time t. Assume ϕ is a uniformly distributed density function U(0 2π). Averaging with respect to ϕ yields the averaging likelihood ratio as follows

$$\overline{\Lambda}(\boldsymbol{r}) = e^{\Psi/2}\frac{1}{2\pi}\int_0^{2\pi}e^{\sqrt{\Psi}r(T)\cos(\phi+\alpha)}\,d\phi$$

$$= e^{-\Psi/2}I_0[\sqrt{\Psi}r(T)] \tag{13}$$

where $I_0(\psi^{1/2}r(T))$ is the modified Bessel function of the first kind and order zero. Therefore the threshold test becomes :

$$I_0[\sqrt{\Psi}\, r(T)] \underset{H_0}{\overset{H_1}{\gtrless}} e^{\Psi/2}\,\overline{\Lambda}(r) \tag{14}$$

where $e^{\Psi/2}\Lambda(r)$ is the operating threshold. The corresponding receiver implementation is shown in Fig 3

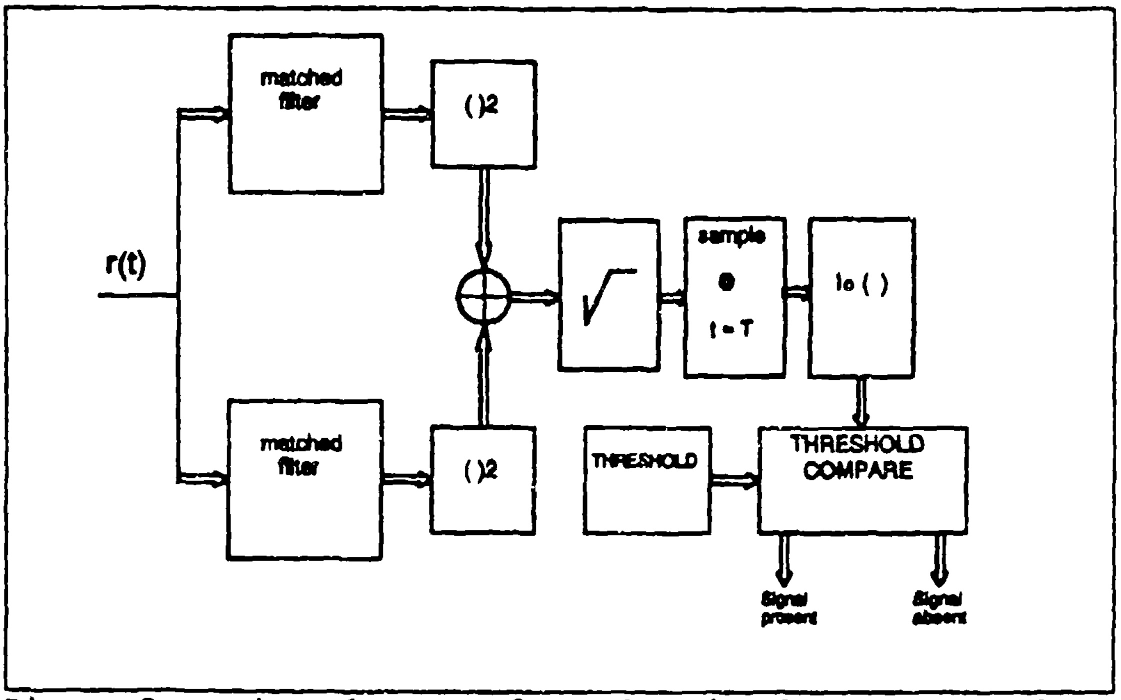

Figure 3: Optimum detector for radar signal of unknown phase

D. OBJECTIVE

The Marcum-Swerling target model is the most commonly used model to calculate radar range performance. The objective of this thesis is to review the underlying theory of radar detection for this model and then develop a MATLAB programs to compute probability of detection and maximum detection range.

II. MODELS AND MATHEMATICAL METHODS

The procedure for the calculation of average probability of detection P_D for Marcum and Swerling cases is as follows:

1) Find the single pulse characteristic function, $C_1(p)$, by transforming the single pulse probability density function $f_{S+N}(y)$.

2) Find the N pulse characteristic function, $C_N(p)=[C_1(p)]^N$.

3) Average the N pulse characteristic function over the target distribution to find the average of $C_N(p)$ ($\bar{C}_N(p)$).

4) Transform $\bar{C}_N(p)$ to find the average ensemble probability density function $f_{s+n}(y)$ and $f_n(y)$ for the N pulse return.

5) Find probability of false alarm P_{fa} by integrating $f_n(y)$ from y_b to infinity.

6) If P_{fa} is given find y_b by means of the mathematical recursive method.

7) Find the average probability of detection P_d, by integrating the density function of signal plus noise $f_{s+N}(y)$ from the threshold y_b to ∞.

A. MARCUM'S (NON-FLUCTUATING) STEADY-STATE TARGET MODEL

The non-fluctuating target model is applied to spherical or nearly spherical objects, such as balloons, many wavelengths in diameter. The target model, therefore, can be represented by a constant-valued radar cross section.

The Rayleigh and Rician probability density function of the envelope detected output for N pulses are given by

$$
\begin{aligned}
f_n(r_i|H_0) &= r_i e^{-r_i^2} \; ; \quad r_i \geq 0 \\
f_{s+n}(r_i|H_1) &= r_i e^{\frac{-(r_i^2+\Psi)}{2}} I_0(r_i\sqrt{\Psi}) \; ; \quad r_i \geq 0
\end{aligned}
\tag{15}
$$

where $\psi = E/N_0$ is the single pulse peak signal-to-noise ratio for a steady target. Since the noise received in the ith observation interval is assumed to be statistically independent of the noise in every other observation interval, and the signals are noncoherent between the different observation interval, the initial phase ϕ_i in Eq(5) in the ith interval is also statistically independent of ϕ_j for all $i \neq j$. The likelihood ratio test in Eq (9) can be written as:

$$
\begin{aligned}
\overline{\Lambda}(\mathbf{r}) &= \frac{f_1(r_1, r_2, \ldots r_N|s_1, s_2 \ldots s_N)}{f_0(r_i|0)} = \prod_{i=1}^{N} \frac{f_1(r_i|s_i)}{f_0(r_i|0)} \\
&= \prod_{i=1}^{N} e^{-\Psi/2} I_0(r_i\sqrt{\Psi}) \\
&= e^{-N\Psi/2} \prod_{i=1}^{N} I_0(r_i\sqrt{\Psi})
\end{aligned}
\tag{16}
$$

From Eq(16), the test statistics is

$$
\sum_{i=1}^{N} \ln I_0(r_i\sqrt{\Psi}) \underset{H_0}{\overset{H_1}{\gtrless}} e^{\Psi/2}\, \overline{\Lambda}(\mathbf{r})
\tag{17}
$$

For small signal, the modified Bessel function $I_0(x)$ can be approximated for $x<1$ as:

$$I_0(x) = 1+\frac{x^2}{4}+\frac{x^4}{64}+.....$$
$$\ln I_0(x) = \ln(1+\frac{x^2}{4}+\frac{x^4}{64}+...) \approx \frac{x^2}{4}+f(x^4) \tag{18}$$

Therefore Eq(17) can be rearranged and modified as

$$\sum_{I=1}^{N} r_i^2 \geq \eta' \tag{19}$$

where η' is the operating threshold which is determined by specifying the false alarm probability.

Marcum utilizes a square law detector law which allows the signal-plus-noise probability density function to be expressed directly in the signal-to-noise ratio ψ. To simplify the calculation, let

$$Y = \sum_{I=1}^{N} \frac{r_i^2}{2} = \sum_{i=1}^{N} q_i \tag{20}$$

which Y is compared to a suitably modified threshold Y_b .

To find the probability density of Y, first the probability density of $f_{s+n}(q_i)$ is found by Jacobain transformation $f_{s+n}(r_i)$ from Eq(15) into $f_{s+n}(q_i)$ as follows:

$$f_{s+n}(q_i) = \frac{f_{S+N}(r_i)}{|J|} = e^{\frac{q_i+\psi}{2}} I_0(\sqrt{2q_i\psi}), \qquad q_i \geq 0 \tag{21}$$

Since random variable Y is the sum of N statistically independent random variables q_i, by applying the relationship between the sum random variable and the individual random variable of the characteristic function, and the Campbell and Foster tables of Fourier transforms ; the $C_Y(p)$ is :

$$C_Y(p) = \prod_{i=1}^{N} C_{q_i}(p) = \prod_{i=1}^{N} \int_{-\infty}^{\infty} e^{jpq_i} f_{S+N}(q_i)\, dq_i$$
$$= \prod_{i=0}^{N} \int_{0}^{\infty} e^{jpq_i}\, e^{-(q_i+\Psi/2)}\, I_0(\sqrt{2q_i\Psi})\, dq_i \tag{22}$$
$$= \frac{e^{-N\Psi/2}}{(1+p)^N}\, e^{\frac{N\Psi}{2(1+p)}}$$

From Campbell and Foster Tables of Fourier transforms pair 650.0, the probability density function of $f_{s+n}(Y)$ can be found by taking the inverse transform of $C_Y(p)$.

$$f_{s+n}(Y) = \left(\frac{2Y}{N\Psi}\right)^{(N-1)/2} e^{-Y-N\Psi/2}\, I_{N-1}(\sqrt{2N\Psi Y}), \qquad Y\geq 0 \tag{23}$$

For small Y, the modified bessel function $I_{N-1}(Y)$ can be approached as:

$$I_m(Y) = \frac{Y^m}{2^m m!}\left[1 + \frac{Y^2}{2^2(m+1)} + \frac{y^4}{2^5(m+1)(m+2)} + \ldots\right] \tag{24}$$

Therefore the resulting normalized square-law detected probability density function is

$$f_{s+n}(Y) = \frac{Y^{N-1} e^{-Y-N\Psi/2}}{(N-1)!}, \qquad Y\geq 0 \tag{25}$$

for noise only the probability of density function $f_n(Y)$ is

$$f_n(Y) = \frac{Y^{N-1}e^{-Y}}{(N-1)!}, \quad Y \geq 0 \tag{26}$$

A computer simulation of $f_n(Y)$ at the square law detector output is given in Figure 4:

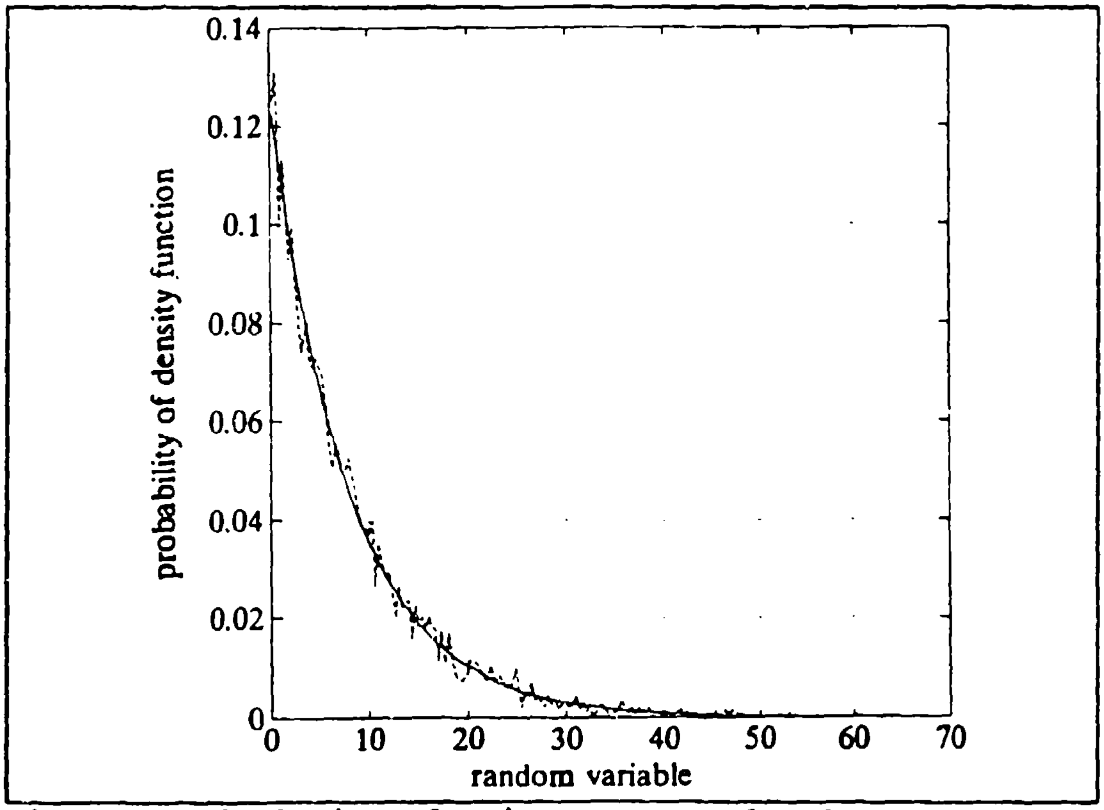

Figure 4: Simulation of noise at suare law detector output

After square-law detection, the normalized square-law detected variate (Y) is summed over the N-pulses in the pulse train and then compared against a normalized threshold voltage (y_b) to determine the presence or absence of a target return.

The probability of false alarm can be obtained by integrating from the threshold (y_b) to infinity

$$P_{fa} = \int_{yb}^{\infty} \frac{Y^{N-1} e^{-y}}{(N-1)!} dY \tag{27}$$

The incomplete Pearson gamma function which is very useful for describing M-S model is defined as:

$$I(u,p) = \int_{0}^{\sqrt{(u(p+1))}} \frac{e^{-\gamma}\gamma^{P}}{p!} d\gamma \tag{28}$$

And in term of the Pearson's incomplete gamma function, Eq(27) is

$$P_{fa}(N, Y_b) = 1 - I[Y_b/N^{1/2}, (N-1)] \tag{29}$$

where $u=(y_b/N^{1/2})$ and $P = N-1$.

Eq(29) can be successively integrated by parts and given as:

$$\begin{aligned} P_{fa}(N, Y_b) &= \frac{y_b^{N-1} e^{-Y_b}}{(N-1)!}\left[1 + \frac{(N-1)}{Y_b} + \frac{(N-1)(N-2)}{Y_b^2} + \ldots\right] \\ &= \sum_{K=0}^{N-1} \frac{y_b^K e^{-Y_b}}{K!} \end{aligned} \tag{30}$$

The above equation also can be approximated for N>>1 by letting

$$\left[1 + \frac{N-1}{Y_b} + \frac{(N-1)(N-2)}{Y_b^2} + \ldots\right] \approx \frac{1}{1 - \frac{N-1}{Y_b}} \qquad N \gg 1 \tag{31}$$

and by applying Stirling's approximation

$$N! = \sqrt{2\pi N}\left(\frac{N}{e}\right)^{N} \tag{32}$$

The probability of false alarm in Eq(30) can be approximated as

$$P_{fa} \approx \sqrt{\frac{N}{2\pi}} \left(\frac{Y_b}{N}\right)^N \frac{e^{-Y_b+N}}{Y-b-N+1} \qquad N \gg 1 \tag{33}$$

The probability of detection for steady-target (case 5) can be obtain by integrating $f_{s+n}(Y)$ from y_b to ∞ and is given by

$$\begin{aligned} P_D(Y) &= \int_{Y_b}^{\infty} f_{s+n}(Y)\ dY \\ &= 1 - \int_0^{Y_b} \left(\frac{2Y}{N\psi}\right)^{(N-1)/2} e^{-Y-(N\psi)/2} I_{N-1}(\sqrt{2N\psi Y}) \qquad Y \geq 0 \end{aligned} \tag{34}$$

The Q function is defined as

$$Q_N(\alpha,\beta) = \int_{\beta}^{\infty} v\left(\frac{v}{\alpha}\right)^{N-1} \exp\left[-\left(\frac{\alpha^2+v^2}{2}\right)\right] I_{N-1}(\alpha v)\, dv \tag{35}$$

For N pulses the probability of detection P_D can be written in terms of Q function as follows:

$$P_D(y) = Q(\sqrt{2N\psi}, \sqrt{2Y_b}) \tag{36}$$

Another approximation can be made by using Gram-Charlier series expansion (Appendix A) and noting that Gaussian family is closed under linear operation.

$$\begin{aligned} m_1 &= (-j)\frac{dC_Y(p)}{dp}\Big|_{p=0} = N(1+\psi/2)\,. \\ m_2 &= (-j)^2\frac{d^2C_Y(p)}{dp^N}\Big|_{p=0} = N^2(1+\psi/2)^2 + N(1+\psi) \\ \sigma^2 &= m_2 - m_1^2 = N(1+\psi) \end{aligned} \tag{37}$$

where m_i is the ith moment of the distribution of Y and σ is the variance of Y.

$$f_{s+n}(Y) = \frac{1}{\sqrt{2\pi N(1+\Psi)}} \frac{e^{-[Y-N(1+\Psi/2)]^2}}{2N(1+\Psi)}$$
$$f_n(Y) = \frac{1}{\sqrt{2\pi N}} e^{(-Y-N)^2/2N} \tag{38}$$

Therefore the probability of false alarm can be represented as:

$$P_{fa} = \int_{Y_b}^{\infty} \frac{1}{\sqrt{2\pi N}} e^{-\frac{(Y-N)^2}{2N}} dY = \int_{\frac{Y_b-N}{\sqrt{N}}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{\frac{-Z^2}{2}} dZ \quad \text{where } Z = \frac{Y_b-N}{\sqrt{N}} \tag{39}$$
$$= \phi[-\frac{Y_b-N}{\sqrt{N}}]$$

Y_b can be solved from the above equation

$$Y_b = \sqrt{N}\,\phi^{-1}(P_{fa}) + N \tag{40}$$

The probability of detection can be obtained as

$$P_D = \phi[\frac{(Y_b-N(1+\Psi/2))}{\sqrt{N(1+\Psi)}}] \tag{41}$$

B. SWERLING'S (FLUCTUATING) TARGET MODELS

1. SWERLING CASES 1 AND 2

The radar cross section of a target (σ) is the area of a hypothetical reflector that scatters all radar beam energy it intercepts omnidirectionally, such that it produces an echo at the radar antenna equal to that energy actually received from the target; that is

$$\sigma = \frac{(\textit{power received by antenna})}{(\textit{incident power density per } 4\pi \textit{ rad})}$$

$$= \lim_{R \to \infty} 4\pi R^2 \frac{|\bar{E}_R|^2}{|\bar{E}_I|^2} \tag{42}$$

Where E_I is the electric field incident on the target, E_R is the reflected field as measured at the radar receiver antenna, and R is the distance from the target to the radar receiver antenna.

The total electric field received from a complex target can be expressed as a summation, namely,

$$E_R = \left| \sum_{K=1}^{n} |\bar{E}_{R_K}| \exp\left(\frac{j4\pi d_K}{\lambda}\right) \right| \tag{43}$$

and therefore the radar cross section is

$$\sigma = \left|\sum_{K=1}^{K=n} \sqrt{(\sigma_K)} \exp\left(\frac{j4\pi d_K}{\lambda}\right)\right|^2$$

$$= \left|\sum_{K=1}^{K=n} \sqrt{(\sigma_K)} \left[\cos\left(\frac{4\pi d_K}{\lambda}\right) + j\sin\left(\frac{4\pi d_K}{\lambda}\right)\right]\right|^2 \tag{44}$$

where

σ_K is the crossing section of individual scattering elements,

n the number of scattering elements,

d_K the range of the Kth element, and

λ the wavelength.

In order to relate the proceeding to a real target two assumptions shall be made:

(1) On a short term basis $4\pi d_k/\lambda$ is a random variable which can take on any value from 0 to 2π with equal probability.

(2) The individual scattering elements have equal radar cross sections, that is, $\sigma k=\sigma_0$.

Therefore the random variables can be expressed as the x component $(\sigma_0)^{(1/2)}\cos(4\pi d_k/\lambda)$ and the y component $(\sigma_0)^{1/2}$ $\cos(4\pi d_k/\lambda)$, the problem of determining the probability density function of σ is identical to the problem of determining the distance moved from an origin in a two dimension walk problem of n steps, where the length of each step is $(\sigma_0)^{1/2}$ and the direction of each step is perfectly random.

The probability of having a component (x,y) after n steps where n is a comparativity large number is given by

$$w(x,y)\ dx\ dy = \frac{\exp[-(x^2+y^2)/n\sigma_0]}{\pi n\sigma_0}\ dx\ dy \qquad \textbf{(45)}$$

Converting from rectangular coordinates to polar coordinates yields

$$w(R,\theta)\ dR\ d\theta = \frac{R}{\pi n\sigma_0}\exp[-(R^2)/n\sigma_0]\ dR\ d\theta \qquad \textbf{(46)}$$

The marginal distribution of R, obtained by integrating the preceeding with respect to θ, is

$$w(p)\ dR = \frac{2R}{n\sigma_0}\exp[-(R^2)/n\sigma_0]\ dR \qquad \textbf{(47)}$$

Now, since $\sigma=R^2=x^2+y^2$ and $d\sigma=2R\ dR$ Eq.(47) becomes

$$w(\sigma) = \frac{\exp[-\sigma/n\sigma_0]}{n\sigma_0} \qquad for \quad \sigma > 0 \tag{48}$$

$w(\sigma)=0$ for $\sigma\leq 0$. This demonstrates that the probability density function of the cross section is exponentially distributed with an average cross section $\bar{\sigma}=n\sigma_0$. Thus, the average total echoing area is the sum of the individual echoing areas of the individual elements.

a. Case 1 Scan-to-scan fluctuations of exponential density function.

The scan-to-scan fluctuations of an exponential nature can be applied to targets such as jet aircraft when a radar having a fairly high pulse repetition rate and scan rate is employed.

For case 1, Swerling derived the probability of detection by initially defining a target model where the received signal power is exponentially distributed, namely,

$$\begin{aligned} w(x,\bar{x}) &= \frac{\exp[-x/\bar{x}]}{\bar{x}} && for \quad x > 0 \\ &= 0 && for \quad x < 0 \end{aligned} \tag{49}$$

where x = input signal-to-noise ratio.

$\bar{x}$ = average of x over all targets fluctuations.

The characteristic equation for the probability density function resulting from the integration of N pulse returns from an exponential fluctuating target, when there is complete correlation between pulses, is derived from the non-fluctuating case (Swerling case 5) characteristic function as follows:

$$C_N = [C_1]^N = \frac{e^{-Nx} e^{\left(\frac{Nx}{P+1}\right)}}{(P+1)^N}$$

$$\overline{C}_N(p) = \int_0^\infty w(x,\overline{x}) \frac{e^{-Nx\left(\frac{p}{p+1}\right)}}{(p+1)^N} dx \tag{50}$$

$$= \frac{1}{(p+1)^{N-1}\,[1+p(1+N\overline{x})]}$$

For noise only, the characteristic equation is the same as for case 5.

$$C_N = \frac{1}{(1+p)^N} \tag{51}$$

The probability density function $f_n(y)$ obtained by using Campbell and Foster 431, p.44, is the same as the probability density function of the nonfluctuating case. i,e,

$$f_n(Y) = \frac{Y^{N-1} e^{-y}}{(1+N)!} \tag{52}$$

For signal plus noise, the probability density function $f_{s+n}(Y)$ was obtained by using Campbell and Foster pair 581.7, p.64. For $N>1$, the density function $f_{s+n}(Y)$ is

$$f_{s+n}(Y)=\int_{-\infty}^{\infty}\frac{1}{(p+1)^{N-1}\ [1+p(1+N\bar{x})]}\ e^{j2\pi py}\ dp$$

$$=\frac{1}{N\bar{x}}\ e^{-y/(1+N\bar{x})}\ \frac{1}{\Gamma(N-1)\ (\frac{N\bar{x}}{1+N\bar{x}})^{N-1}}\ [\Gamma(N-1)-\Gamma(N-1,\frac{N\bar{x}}{1+N\bar{x}}y)]$$

$$=\frac{1}{N\bar{x}}\ e^{-y/(1+N\bar{x})}\ (1+\frac{1}{N\bar{x}})^{N-2}(1-\frac{\Gamma(N-1,\frac{N\bar{x}}{1+N\bar{x}}y)}{\Gamma(N-1)}) \tag{53}$$

$$=\frac{1}{N\bar{x}}\ e^{-y/(1+N\bar{x})}\ (1+\frac{1}{N\bar{x}})^{N-2}I(\frac{y}{1+\frac{1}{N\bar{x}}\ \sqrt{N-1}},N-2)$$

where $\Gamma(v,z)$ is the incomplete gamma function and $I(u,p)$ is the incomplete Pearson gamma function.

For $N=1$, the probability density function can be obtained from Campbell and Foster pair 438, p.45 as

$$f_{s+n}(Y)=e^{(-Y/1+\bar{x})}/(1+\bar{x}) \tag{54}$$

therefore the probability of detection can be obtained by integrating Eq(54) from 0 to Y_b and the result is given by Swerling as:

$$1-P_D=\int_0^{Y_b} f_{s+n}(y)\ dv=I[\frac{Y_b}{\sqrt{(N-1)}},N-2]-(1+N\bar{x})\,f(Y_b) \tag{55}$$

From above, P_D can be obtained as

$$P_D=1-I[\frac{Y_b}{\sqrt{N-1}},(N-2)]+(1+\frac{1}{N\bar{x}})^{N-1}$$
$$I(\frac{Y_b}{[1+(1/N\bar{x})\sqrt{N-1}]},N-2)\,e^{-Y_b/(1+N\bar{x})} \tag{56}$$

Another approximation can be made by inspecting Eq(56) and let $N\bar{x}>>1$ and $P_{fa}<<1$ such that the Pearson incomplete gamma functions are close to unity. Then Eq(56) can be rewritten approximately as

$$P_D \simeq \left(1+\frac{1}{N\bar{x}/2}\right)^{N-1} e^{\frac{-Y_b}{1+N\bar{x}/2}} \qquad \begin{matrix} N\bar{x}>1 \\ P_{fa}<1 \end{matrix} \tag{57}$$

Taking the logarithm of Eq(57) and using a series expansion of each term,

$$\begin{aligned} \ln P_d &\simeq (N-1)\ln\left(1+\frac{1}{N\bar{x}}\right) - \frac{Y_b}{(N\bar{x})\left(1+\frac{1}{N\bar{x}}\right)} - \\ &\simeq (N-1)\left[\frac{1}{(N\bar{x})} - \frac{1}{2}\frac{1}{(N\bar{x}/2)^2} - \frac{1}{3}\frac{1}{(N\bar{x}/2)^3} - \ldots\right] \\ &\quad - Y_b\left[\frac{1}{N\bar{x}/2} - \frac{1}{(N\bar{x}/2)^2} + \frac{1}{(N\bar{x}/2)^3} - \ldots\right] \end{aligned} \tag{58}$$

Taking only the first terms and using Eq(40) yields

$$\begin{aligned} \ln p_D &\simeq -\frac{2}{N\bar{x}}(Y_b - N + 1) \\ &\simeq -\frac{2}{N\bar{x}}\left[\sqrt{N}\,\Phi^{-1}(P_{fa}) + 1\right] \end{aligned} \tag{59}$$

b. Case 2 Pulse-to-pulse fluctuations of exponential density function.

Pulse-to-pulse fluctuations of an exponential nature apply to targets such as propeller-driven aircraft (if the propellers contribute a significant portions of the echo area), to targets where small changes in orientation would establish significant changes in echoing area (such as long thin subjected to a high-frequency signal), or to targets viewed by radars with sufficiently low repetition rates.

For case two, the received signal return still belongs to exponential distribution. For this case the signal is completely decorrelated (pulse to pulse fluctuations exist), the probability density function for noise only (signal-to-noise ratio equals to 0) still the same with Case 1. The probability of density function of signal plus noise for case 2 can be obtained with the same procedures, but have the pulse to pulse integration. The characteristic equation for the single pulse is obtained by letting the characteristic equation of case 1 to be N=1

$$\overline{C}_1(p) = \frac{1}{[1+p(1+\overline{x})]} \tag{60}$$

For N pulses which is completely decorrelated (independent), the characteristic equation is just the Nth power of single pulse

$$\overline{C}_N(p) = [\overline{C}_1(p)]^N = [\frac{1}{1+p(1+\overline{x})}]^N \tag{61}$$

Therefore the corresponding probability of density function obtained by using Campbell and Foster *"Tables of Fourier transforms"* pair 431, p.44 yield:

$$f_{s+n}(y) = \int_{-\infty}^{\infty} [\frac{1}{1+p(1+\overline{x})}]^N e^{2\pi py} dy = \frac{y^{(N-1)} e^{(-y/(1+\overline{x}))}}{(1+\overline{x})^N (N-1)!} \quad \textbf{(62)}$$

The probability of detection can be expressed as:

$$\begin{aligned} P_D &= 1-\int_{Y_b}^{\infty} f_{s+n}(y)\ dy \\ &= 1-\frac{1}{(N-1)!}\int_0^{Y_b} \frac{y}{1+\overline{x}}^{N-1} e^{-y/(1+\overline{x})} \frac{dy}{1+\overline{x}} \\ &= 1-I[\frac{Y_b}{(1+\overline{x})\sqrt{N}}, (N-1)] \end{aligned} \quad \textbf{(63)}$$

Another approximation can be made by applying that the Gaussian family is closed under linear operation and by using Gram-Charlier series expansion (Appendix A) so that

$$\begin{aligned} m_1 &= (-j)\frac{dC_Y(p)}{dp}\Big|_{p=0} = N(1+\overline{x}) . \\ m_2 &= (-j)^2 \frac{d^2 C_Y(p)}{dp^N}\Big|_{p=0} = N(N+1)(1+\overline{x})^2 . \\ Variance &: \sigma^2 = m_2 - m_1^2 = N(1+\overline{x})^2 \end{aligned} \quad \textbf{(64)}$$

The approximate probability density function then is

$$\begin{aligned} f_{s+n}(Y) &= \frac{1}{\sqrt{2\pi N}(1+\overline{x})} e^{\frac{-[Y-N(1+\overline{x})]^2}{2N(1+\overline{x})^2}} \\ f_N(Y) &= \frac{1}{\sqrt{2\pi N}} e^{(-Y-N)^2/2N} \end{aligned} \quad \textbf{(65)}$$

Therefore the probability of detection and false alarm can be obtained by integrating from the threshold level y_b to ∞ and are represented as:

$$P_D = \Phi\left[\frac{(Y_b - N(1+\bar{x}))}{\sqrt{N}(1+\bar{x})}\right] \quad ; \quad P_{fa} = \Phi\left[\frac{(Y_b - N)}{\sqrt{N}}\right] \tag{66}$$

2. SWERLING'S CASE 3 AND 4

For Swerling's case 3 and case 4, the radar cross section can be described by chi-square distributions with four degrees of freedom. The density function is commonly associated with tabilized missile tankage and can be expressed as:

$$w(x,\bar{x}) = \frac{4x}{\bar{x}^2}\, e^{-\left(\frac{2x}{\bar{x}}\right)} \qquad x>0 \tag{67}$$

a. Case 3 Scan-to-scan fluctuations with a chi-square density function with four degrees of freedom.

The characteristic equation for case 3 which represents the condition of complete correlation (scan-to-scan, no pulse-to-pulse fluctuation) is given by :

$$\bar{C}(p) = \int_{-\infty}^{\infty} \frac{4x}{\bar{x}^2}\, e^{-\left(\frac{2x}{\bar{x}}\right)}\, \frac{e^{-Nx\left(\frac{p}{p+1}\right)}}{(p+1)^N}\, dx = \frac{(p+1)^{2-N}}{\left[1+p\left(1+\frac{N\bar{x}}{2}\right)\right]^2} \tag{68}$$

The characteristic equation and the probability of density function for noise only remain the same as in case 1 and 2

$$\bar{C}_N(p) = \frac{1}{(p+1)^N}$$
$$f_n(y) = \int_{-\infty}^{\infty} \frac{1}{(p+1)^N}\, e^{j2\pi py}dp = \frac{y^{N-1}\, e^{-y}}{(N-1)!} \tag{69}$$

For signal plus noise the probability of density function can be expressed as:

$$
\begin{aligned}
f_{s+n}(y) &= \int_{-\infty}^{\infty} \frac{(p+1)^{2-N}}{[1+p(1+\frac{N\bar{a}}{2})]^2}\, e^{j2\pi py}\, dp \\
&= \frac{y^{N-1}\, e^{-Y}}{(N-1)!\, [(1+(N\bar{x}/2)]^2}\, F_1[2, N, \frac{y}{1+(2/N\bar{x})}]
\end{aligned}
\tag{70}
$$

where the density function comes from Fourier transform pair 581.1 in the Campbell and Foster tables given as

$$
\frac{1}{(s+\rho)^{\alpha+\nu}(s+\sigma)^{\alpha-\nu}} \leftrightarrow \frac{1}{\Gamma(2\alpha)\,(\rho-\sigma)^{\alpha}}\, Y^{\alpha-1}\, e^{-(\rho\sigma)Y/2}\, x \quad M_{\nu,\alpha-1/2}[(\rho-\sigma)Y] \quad , Y<0 \tag{71}
$$

where

$$
M_{\nu,\mu}(Z) = Z^{\mu+\frac{1}{2}}\, e^{-\frac{Z}{2}}\, F_1(\mu-\nu+\frac{1}{2}, 2\mu+1; Z) \tag{72}
$$

Eq(72) can be simplified using the relationships for the confluent hypergeometric function, that is

$$
\begin{aligned}
F_1(2,N;Z) &= (Z+2-N)\, F_1(1,N;Z) + N-1 \\
F_1(1,N;Z) &= e^{Z}\, Z^{-N+1}(N-1)!\; I[\frac{Z}{\sqrt{(N-1)}}, N-2]
\end{aligned}
\tag{73}
$$

Swerling uses two identities relating to confluent hypergeometric functions in order to expand the proceeding into more familiar Pearson's form of the incomplete gamma function, $I(u,p)$ which is defined in Eq (30), thus Eq (73) becomes

$$f_{s+n}(y)=\frac{[1+(2/N\bar{x})]^{N-2}y}{[1+(N\bar{x}/2)]^2}\,I[\frac{y}{[1+2/(N\bar{x})]\sqrt{N-1}},N-2]\;e^{(-y/1+(N\bar{x}/2))}$$

$$-\frac{(N-2)\,[1+(2/(N\bar{x})]^{N-1}}{[1+(N\bar{x}/2)]^2}\,I[\frac{y}{[1+(2/N\bar{x})]\sqrt{N-1}},N-2] \qquad (74)$$

$$e^{-y/(1+N\bar{x})}+\frac{y^{N-1}e^{-y}}{[1+(N\bar{x}/2)]\,(N-1)\,!}$$

The probability of detection can be obtained by integrating the density function from the threshold Y_b to ∞ as:

$$P_D=\frac{Y_b^{N-2}e^{-Y_b}}{(N-2)\,!}\,\frac{2Y_b}{N\bar{x}+2}+\sum_{m=0}^{N-2}e^{-Y_b}\frac{Y_b^m}{m!}+(\frac{N\bar{x}+2}{N\bar{x}})^{N-2}e^{\frac{-2Y_b}{N\bar{x}+2}}$$

$$[1-\frac{2(N-2)}{N\bar{x}}+\frac{2Y_b}{N\bar{x}+2}]\,[1-\sum_{m=0}^{N-2}e^{\frac{-2Y_b}{2+N\bar{x}}}\frac{(\frac{2Y_b}{2+N\bar{x}})^m}{m!})]\,;\;N\geq 2 \qquad (75)$$

b. Case 4 pulse-to-pulse fluctuations with a chi-square density function with four degrees of freedom

With no correlation, the characteristic equation for a single hit is obtained by letting N=1 in Eq(52)

$$\bar{C}(p)=\frac{1+p}{[1+p(1+\frac{\bar{x}}{2})]^2} \qquad (76)$$

The characteristic equation for the sum of N pulse can be obtained by the Nth power of a single hit

$$\overline{C}_N(p) = [\overline{C}_1(p)]^N = \frac{(1+p)^N}{1+p(1+\overline{x}/2)]^{2N}} \tag{77}$$

The inverse Fourier transform of above equation is obtained from Campbell and Foster transform Pair 581.1,p.64, this yields the following expression for the probability of density function in the condition of signal plus noise

$$\begin{aligned} f_{s+n}(y) &= \int_{Y_b}^{\infty} \frac{4x}{\overline{x}^2} e^{-\left(\frac{2x}{\overline{x}}\right)} \frac{(1+p)^N}{[1+p(1+\overline{x}/2)]^{2N}} \\ &= \frac{y^{N-1} e^{\frac{-y}{1+\overline{x}/2}}}{(1+\overline{x})^{2N}} (N-1)!\ F_1\left(-N, N; \frac{-\overline{x}/2}{1+\overline{x}/2} y\right) \end{aligned} \tag{78}$$

The confluent hypergeometric function $F_1(-N,N;a)$ can be expanded as:

$$\begin{aligned} F_1(-N,N;a) &= 1 + \frac{-N}{N}\frac{a}{1!} + \frac{(-N)(-N+1)}{N(N+1)}\frac{a^2}{2!} + \frac{(-N)(-N+1)(-N+2)}{N(N+1)(N+2)}\frac{a^3}{3!} + \ldots\ldots. \\ &= \sum_{K=0}^{N} \frac{(-1)^K[N!/(N-K)!]}{[(N+K-1)!/(N-1)!]} \frac{a^K}{K!} \end{aligned} \tag{79}$$

From Eq(78) and Eq(79), the density function $f_{s+n}(Y)$ can be rewritten as

$$f_{s+n} = \frac{y^{N-1} e^{-\frac{y}{1+\overline{x}/2}} N!}{(1+\overline{x}/2)^{2N}} \sum_{K=1}^{N} \left(\frac{\overline{x}/2}{1+\overline{x}/2}\right)^K \frac{y^K}{[(N+K-1)!\ (N-K)!\ K!]} \tag{80}$$

The probability of detection is obtained by integrating Eq(80) from the desired threshold Y_b to ∞. From the definition of incomplete gamma function, the probability of detection can be written as:

$$P_D = \int_{Y_b}^{\infty} f_{s+n}(y)\ dy$$

$$= 1-\frac{N!}{(1+\overline{x}/2)^N}\sum_{K=0}^{N}\left(\frac{\overline{x}}{2}\right)^K \frac{I\left[\frac{Y_b}{(1+\overline{x}/2)\sqrt{N+K}}, N+K-1\right]}{K!\ (N-K)!} \tag{81}$$

With Eq(40), the probability of detection can be approximated as:

$$P_D \approx 1-\frac{N!}{(1+\overline{\sigma}/2)^N}\sum_{K=0}^{N}\left(\frac{\overline{\sigma}}{2}\right)^K \frac{I\left[\frac{\sqrt{N}\,\Phi^{-1}(P_{fa})+N}{(1+\overline{\sigma}/2)\sqrt{N+K}}, N+K-1\right]}{K!\ (N-K)!} \tag{82}$$

C. SEARCH RADAR DETECTION RANGE CALCULATION

The Marcum-Swerling theory represented by extensive sets of curves from the computer programs can be used to determine the detection range of a practical radar by introducing detection loss and others parameters. There are many types of detection losses which have been identified so far, and when these are considered, reasonable predictions of radar performance can be obtained.

The radar's detection range can be determined by applying the desired signal-to-noise ratio determined from the Marcum-Swerling theory and the calculated detection loss, as given by

$$R_{max} = [\frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 kTBF_n L(\frac{S}{N})}]^{1/4} \quad (m) \tag{83}$$

where

R_{max} = The maximum detection range in meters

P_t = Peak transmitter power in Watts

λ = Wavelength in meters

$G_{t,r}$ = Transmitter and receiver antenna power gains

σ = Average radar cross section in square meters

k = Boltzmann's constant= 1.38×10^{-23} J/deg.

T = Effective system input noise temperature in degrees Kelvin (^{0} K)

B = Receiver bandwidth in Hertz.

L = Detection system power loss factor

F_n= The receiver noise figure

S/N= The smallest output signal-to-noise ratio

When n pulse are integrated previous equation can be written as

$$R_{max} = [\frac{P_t G_t G_r \lambda^2 \sigma n}{(4\pi)^3 kTBF_n L(\frac{S}{N})_n}]^{1/4} \quad (m) \tag{84}$$

where the parameters are the same as that of Eq (83) except that $(S/N)_n$ is the signal-to-noise ratio of one of the n pulses that are integrated to produce the required probability of detection for a specified probability of false alarm.

III. SOFTWARE DEVELOPMENT

This chapter describes the development of MATLAB programs for the efficient and accurate computation of probability of detection based on Marcum and Swerling theory of radar detection. The MATLAB source code is given in Appendix B, and complete programs are available from Professor G.S. Gill, Code EC/Gl, Naval Postgraduate School, Monterey, CA 93943.

A. PROGRAM STRUCTURE

The overall program structure is shown in Figure (5). The structure is that of a main menu program which calls various submenu programs (mscurve.m number.m number.m) as required. The submenu programs, when called, will then display the purpose of the subprograms. When called, the subprograms will call the function programs to do the actual computation. It is possible to exit the process from either the main program or from the subprograms or the function programs. The advantage of this format is that if the user wants to change one of the subprograms or function programs and wants to add other programs, this system can accommodate it. For each subprogram there is an error detect prevention and data entry double check function to inform the user and restart the process.

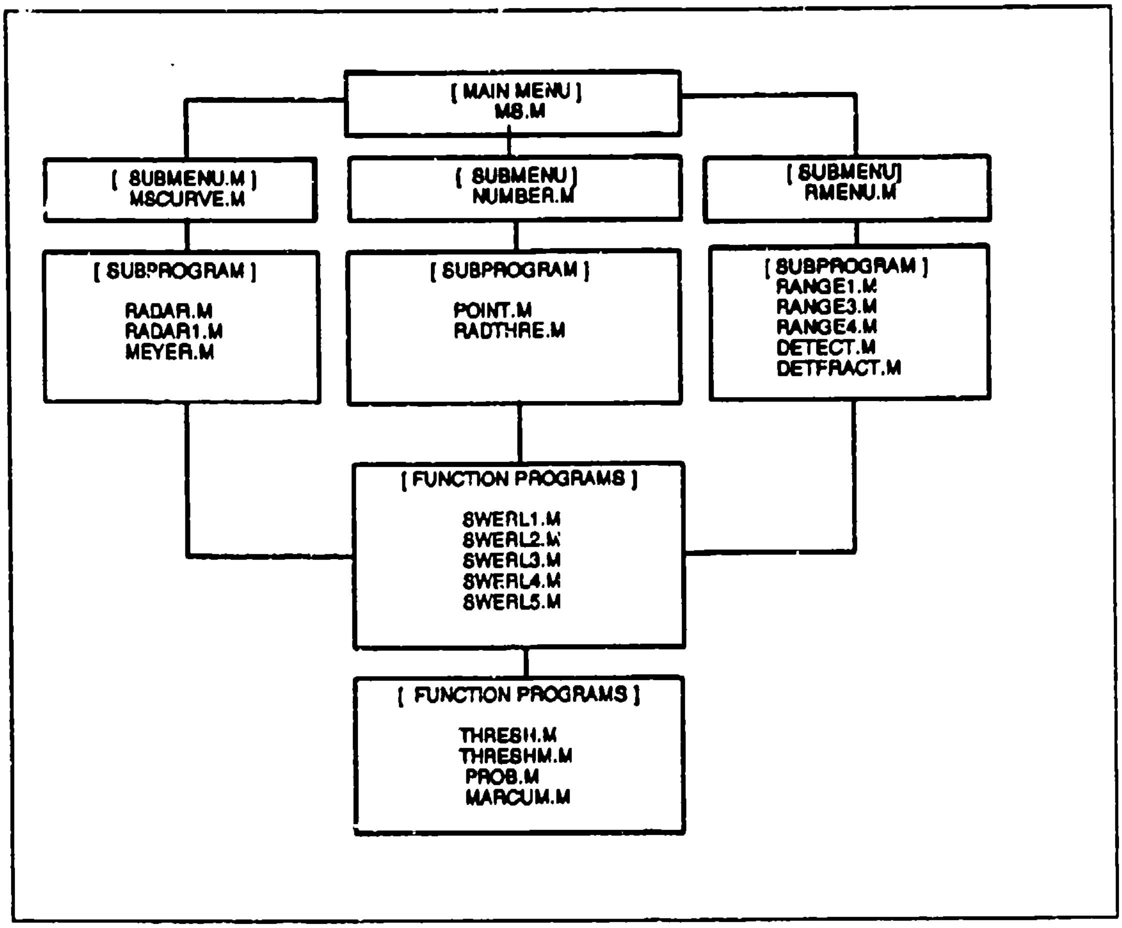

Figure 5: Program structure

1. User's guide and instruction

To use these programs, the following MATLAB files have to be copied to the user subdirectory. A brief explanation of each is also given.

•ms.m is the main menu program and gives brief descriptions of the M-S model and displays the main menu.

•mscurve.m ,rmenu.m, and number.m are the submenu programs. These give the purpose of the programs when called.

•radar.m, radral.m, meyer.m, are the subprogram used to integrate swerl1.m, swerl2.m, swrl3.m, swerl4.m, swerl5.m function programs to calculate and plot the curves p_d vs S/N, p_d vs p_{fa} and p_d vs N, respectively.

•range1.m, range3.m, range4.m, are the subprograms responsible for integrating and calculating the detection range. The user inputs the specified detection loss from detect.m. The range1.m, range3.m, range4.m, will load the data from detect.m and calculate automatically and display the detection curves on the screen.

•point.m radthre.m, are the subprograms responsible for calculating and displaying the numerical results from the function program radpoint.m and marcum.m, respectively.

•swerl1, swerl2, swerl3, swerl4, swerl5, prob.m, thresh.m, threshm.m, bound.m, radpoint.m, noise.m, signal.m, are all function programs responsible for calculating the data from the subprogram and then return the numerical value.

A 386 or 486 personal computer is suggested for greater speed. To start the program type **ms** and press [enter] at the MATLAB prompt.

2. Printing graphical outputs

Graphics output from all programs will be stored as meta files automatically. The operator can print out the desired graphics from MATLAB by typing **!gpp** <filename> [enter]. Once the user restarts the programs, the previous meta files will be deleted automatically.

3. Programs options

Once the ms.m command is given, the first screen seen by the user is as shown in Figure 6.

```
%THIS PROGRAM IS DESIGNED TO ALLOW THE STUDENT TO
%VARY THE PARAMETERS OF THE VARIOUS SWERLING CASES
%IN ORDER TO STUDY THE EFFECTS.
%
% CASE #:     DESCRIPTION

% 1. Returned pulses are of a constant amplitude
%    over one scan, but are uncorrelated from
%    scan to scan.
%
% 2. Returned pulses are uncorrelated from pulse
%    to pulse and correlated  from scan to scan.
%
% 3. Returned pulses are of a constant amplitude
%    over one scan, but are uncorrelated from scan
%    to scan.
%
% 4. Returned pulses are uncorrelated from pulse to
%    pulse and correlated from scan to scan.
%
% 5. The static case with constant S/N and pulse
%    amplitude
```

Figure 6: Main menu descriptions

This screen gives a brief description of the five target models. After pressing the "enter" key, the user will see a second screen as shown at Figure 7.

```
----- MAIN MENU -----

        1) THE M-S CURVES
        2) NUMERICAL DETECTION PROBABILITY CALCULATION
        3) RANGE DETECTION CURVES


    Select a menu number:
```

Figure 7: Main menu

At this point the operator has different choices to make depending upon what he/she wants to do. Once the operator chooses one of the items, the main menu program will transfer to the selected submenu in Figure 8.

```
----- SUB MENU -- [THE M-S CURVES ANALYSIS] -----

        1) PROBABILITY OF DETECTION vs.  S/N
        2) PROBABILITY OF DETECTION vs.  Pfa
        3) PROBABILITY OF DETECTION vs.   N
        4) COMPARSION OF M-S CURVES


    Select a menu number:
```

Figure 8: Submenu Screen

From the submenu the user will have another set of choices. He can choose the item he wants to study. After he chooses one of the items, the following selected screen will be seen in Figure 9, Figure 10, Figure 11 and Figure 12. The user can follow the instructions on the screen to key in the arguments he wants to study. Then a data input screen seen at Figure 13 will display the data for double check. After the completion of above procedure, a selected graphic output will appear on the screen as in Figure 14. After the graphics display, the user can press 'enter' to get the next screen as shown in Figure 15. At this point, the user can either choose to go back to the main menu or exit to print out the graphic display. A selected second submenu and third submenu are shown in Figure 16 and Figure 17; users can follow the same procedure to choose the items they want to study.

```
%***********************************************************
%      THIS PROGRAM RETURNS THE PLOTS FOR THE NUMBER
% OF PULSES AND SWERLING CASE SPECIFIED IN THE PARAMETERS.
% THE PLOTS WILL BE STORED IN METAFILES UNDER THE NAME
% "RADAR.MET" FOR AN EASY PRINT OUT.
%
%  (A)   the swerling case number has to be determined now
%***********************************************************
echo off

 Enter the case number you want to study
```

Figure 9: Subprogram Descriptions Screen (a)

```
************************************************************
  (B).   The number of radar pulses the program is to
         integrate needs to be an integer between 1 and 600
************************************************************
cho off

Number of Pulses to be integrated is n = 10
```

Figure 10: Subprogram Descriptions Screen (b)

```
****************************************************************************
(C).   The probability of false alarm rate curves (pfa) to be plotted
must now be determined.  Each choice of a pfa will result in a different
curve being plotted on the graph.  You need to choose the following;

     1.       The smallest pfa curve to be plotted, pfamin = ?

     2.       The largest pfa curve to be plotted, pfamax = ?

     3.       The step size between pfamin aand pfamax, pfastep = ?

     If you wish to plot only one curve then enter the same value for
     pfamin and pfamax.

     The suggested default step size to use is that of PFASTEP = 10,
     which is quite sufficient.  It is suggested that pfamin and pfamax
     be powers of 10 as that is the normal choice.
****************************************************************************
```

Figure 11: Subprogram Descriptions Screen (c)

```
*************************************************************
  The signal to noise ratio (S/N) in dB for which you wish to
  plot needs to be determined.  The choices you need
  to make are;

1.       The smallest S/N point to be plotted, sdbmin = ?

2.       The largest S/N point to be plotted, sdbmax = ?

         Remember that S/N must be entered in dB.

3.       The stepsize between sdbmin and sdbmax = ?
**************************************************
```

Figure 12: Subprogram Descriptions Screen (d)

```
%-----------------------------------------------------------------
% CHECK YOUR PARAMETERS !
% ---------------------------------------------------------------
echo off
The case number you is     1.00
The number of pulses you choice are     1.00
The max false alarm probability you choice is 1.00e-12
The min false alarm probability you choice are1.00e-12
The pfa stepsize is    10.00
The max signal-to-noise ratio you choice is    10.00
The min signal-to-noise ratio   -10.00
The S/N stepsize is     1.00
%----------------------------------------------------------------
%IF THE PARAMETERS ARE CORRECT PRESS 1
%IF THE PARAMETERS ARE NOT  CORRECT PRESS 2
%----------------------------------------------------------------
```

Figure 13: Input Argument Checking Screen

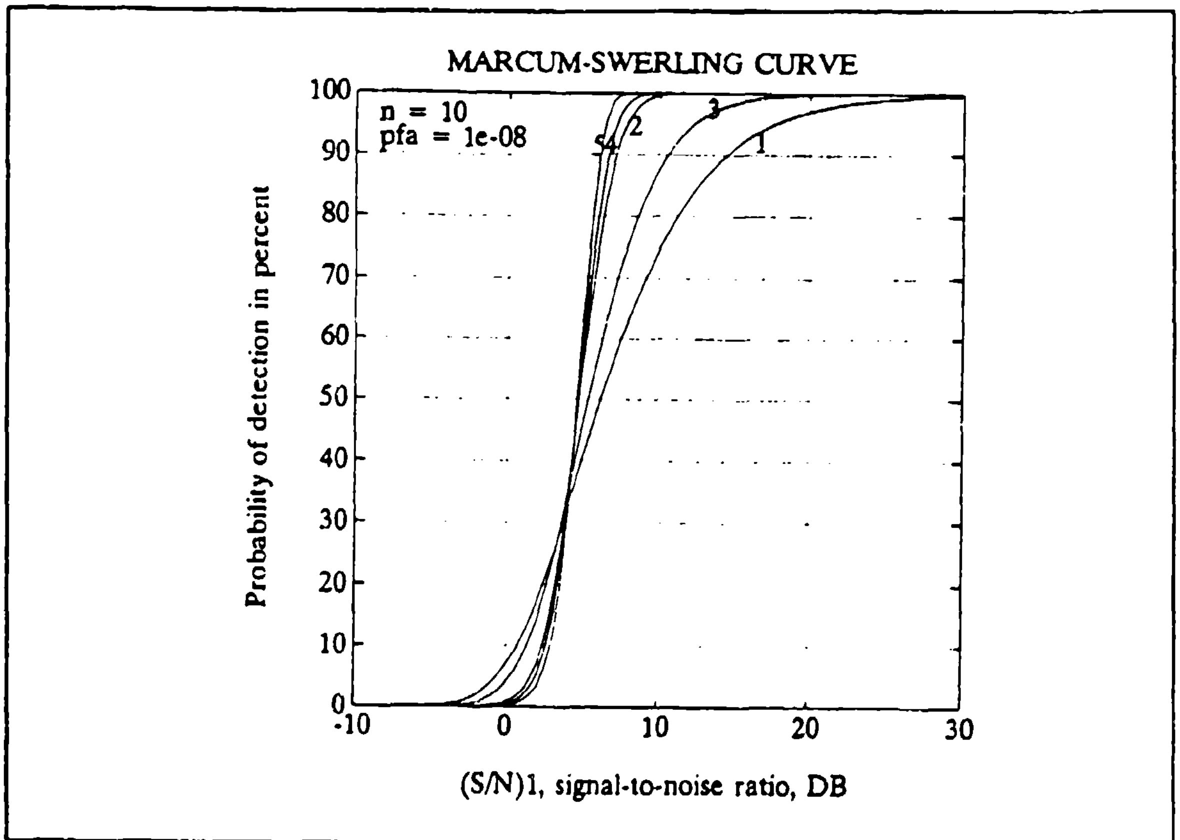

Figure 14: Selected Result

```
If you want to go to the same submenu     ENTER CHOICE = 1

        or

If you want to go to the main menu        ENTER CHOICE = 2

        or

To exit this whole program                  PRESS RETURN
```

Figure 15: Selected menu

```
        ----- SUB  MENU [ NUMERICAL CALCULATION] -----

              1) CALCULATION OF DETECTION PROBABILITY
              2) CALCULATION OF THRESHOLD LEVEL
```

Figure 16: Selected second menu

```
SUB MENU -- [THE DETECTION RANGE ANALYSIS] -----

1) RANGE DETECTION CURVES WITH DIFFERENT PROBABILITY OF FALSE ALARM
2) RANGE DETECTION CURVES COMPARSION
3) RANGE DETECTION CURVES USE THE DETECABILITY FACTOR
```

Figure 17: Selected third menu

B. Algorithms for M-S models

The function programs in Figure 5 are based on the Marcum and Swerling detection theory.

1. Function program algorithms

The function programs can be executed independently as a normal MATLAB function. The input arguments and output data are listed in Table [1] at the end of this chapter. All these programs were implemented with the help feature of the MATLAB environment. Typing "help < name of the function >" will explain how to use them independently.

a. Threshold computation

The detection programs start with the calculation of threshold y_b by applying the equation (30) in Chapter 2.

$$P_{fa}=P_{fa}(N,Y_b)=1-I\left(\frac{Y_b}{\sqrt{N}},N-1\right)=\sum_{K=0}^{N-1}\frac{Y_b^k}{K!}e^{-Y_b} \qquad (85)$$

The false-alarm probability can be represented as a function of N and y_b. To compute probability of detection for all the five cases Y_b is required for a given p_{fa}.

Since Eq(85) is a finite power series, in order to find Y_b for given P_{fa}, a recursive computation method has to be used rewriting Eq(85)

$$P_{fa}(N,Y_b)=\sum_{k=0}^{N-1}\frac{Y_b^k}{K!}e^{-Y_b}=P_{fa}(N-1,Y_b)+\frac{Y_b^{N-1}}{N-1}e^{(-Y_b)}$$
$$=P_{fa}(N-1,Y_b)+L(N-1) \tag{86}$$

$$P_{fa}(N+1,Y_b)=\sum_{k=0}^{N}\frac{Y_b^k}{K!}e^{-Y_b}=P_{fa}(N,Y_b)+\frac{Y_b^N}{N}e^{-Y_b}$$
$$=P_{fa}(N,Y_b)+L(N) \tag{87}$$
$$=P_{fa}(N,Y_b)+L(N-1)\frac{Y_b}{N}$$

From Eq (86) and (87)

$$L(n)=L(N-1)\frac{Y_b}{N} \tag{88}$$

For a single hit (N=1) the false alarm probability and the relation with the following term (N=2) are

$$P_{fa}(1,Y_b)=e^{Y_b} \quad , \quad L(1)=Y_b e^{-Y_b} \tag{89}$$

Therefore this relation allow each term of the expansion to be based on the value of the proceeding term, therefore an algorithm can be formed to compute the values of the detection threshold y_b and the number of integrated pulses N.

The first procedure employed in the algorithm is to define an empirical threshold level $Y_{0,}$ then compute $P(N,y_0)$. On the basis of this empirically determined value y_0, an empirical suggested Y_b is given by

$$Y_b = N - \sqrt{N} + 2.3\sqrt{L}(\sqrt{L} + \sqrt{N} - 1), \quad where \;\; L = -\log P_{fa} \tag{90}$$

This value is used as the starting point to compute $P(N, y_0)$ then compare it against the desired value of P_{fa} and the difference between Y_N and Y_{N+1}. This value of correction Δy can be calculated by using Newton-Raphson method by noting hat

$$Y_{N+1} = Y_N + \frac{\ln \dfrac{P_{fa}(N, Y_N)}{P_{fa}}}{\dfrac{e^{-Y_N} Y_N^{N-1}/(N-1)!}{P_{fa}(N, Y_N)}} \tag{91}$$

The procedure is repeated until the correction magnitude $\Delta y/(\Delta y + y)$ is indicated that Y_b is within a sufficient accuracy. A computer independent algorithm notation is given on Figure 18. In Figure 18, the arrow implies a specification. The normal execution of statements is carried out line by line, starting at the top, but a branch may be designated by an arrow which results from the execution of a statement. A conditional branch is denoted by a colon statement, and the branch is executed if the comparison condition specified on the arrow is satisfied. Otherwise, the next statement in the sequence is executed. Notice in figure 18 that the program is terminated when the value $\Delta y/(\Delta y + y)$ is less or equal to ε. The value of ε can be assigned to be 10^{-6} or 10^{-12}. This accuracy should be sufficient for application.

$L \leftarrow -\log_{10}(p_{fa})$

$y_0 \leftarrow n - -\sqrt{n} + 2.3\sqrt{L}(\sqrt{L} + \sqrt{n} - 1)$

$y \leftarrow y_0$

$Ym \leftarrow \exp(-y)$

$Yms \leftarrow Ym$

$M \leftarrow 0$

$M \leftarrow M+1$

$= \quad N \;:\; M$

$Ym \leftarrow Ym * Y/M$

$Yms \leftarrow Yms + Ym$

$p \leftarrow Yms$

$\Delta Y \leftarrow (p/Ym) * \ln(P/P_{fa})$

$y \leftarrow Y + \Delta Y$

$\leq \quad \epsilon \;:\; \frac{|\Delta Y|}{Y}$

Exit Y_b

Figure 18: Algorithmic Program for Detection Thresholds, Y_b

The MATLAB function for executing this process is named **prob.m and thresh.m** respectively. In Figure 18 ε is the smallest acceptable tolerance value. $\Delta y/(\Delta y+y)$ is compared with ε, and when it is less than or equal to ε, the computation will stop and return the value of false alarm probability. In **thresh.m** the smallest tolerance value was set to be 10^{-6}, this value is sufficient for desired accuracy.

This recursive method will be used in all Swerling cases to find the threshold level y_b.

b. *Swerling case 1 algorithm*

Following equation is used to compute probability of detection for case 1.

$$P_{D_1} = 1 - I\left[\frac{Y_b}{\sqrt{N-1}}, (N-2)\right] + \left(1+\frac{1}{N\bar{x}}\right)^{N-1} I\left(\frac{Y_b}{[1+(1/N\bar{x})\sqrt{N-1}]}, N-2\right) e^{\frac{-Y_b}{(1+N\bar{x})}}$$

$$= P_{fa}(N-1, Y_b) \cdot \left(1+\frac{1}{N\bar{x}}\right)^{N-1} \left[1 - P_{fa}\left(N-1, \frac{Y_b}{1+\frac{1}{N\bar{x}}}\right)\right] e^{\frac{-Y_b}{(1+N\bar{x})}} \tag{92}$$

It is obvious to see that for *N=1* the detection probability is $e^{-(Yb\,/\,1+Nx)}$. The algorithm in computer independent notation which uses equation (92) to compute P_{D1} is shown on Figure 19. Notice in Figure 18 that the detection threshold (Y_b) is independent of the target fluctuation characteristics so that the algorithm given in Figure 18 is used to determine Y_b for all target types. A MATLAB source code to compute probability of detection of case 1 targets is given in Appendix B. The name of this file is swerl1.m.

Enter Y_b, N, X

XN	$\leftarrow$	N * X
Y	$\leftarrow$	Y_b
N	:	1
A	$\leftarrow$	0
YM	$\leftarrow$	EXP (-Y)
YMS	$\leftarrow$	YM
M	$\leftarrow$	0
M	$\leftarrow$	M + 1
N - 1 :		M
YM	$\leftarrow$	YM ∘ Y/M
YMS	$\leftarrow$	YMS + YM
P_2	$\leftarrow$	YMS
1	:	A
A	$\leftarrow$	1
P_1	$\leftarrow$	P2
Y	$\leftarrow$	$Y/(1 + 1/X_N)$
A_1	$\leftarrow$	$(1 + 1/X_N)^{N-1}$
A_2	$\leftarrow$	$EXP\ (-Yb/(1 + X_N))$
P_3	$\leftarrow$	$A1 * A_2 \circ (1 - P_2)$
P	$\leftarrow$	$P_1 + P_3$ ------------> Exit P_{D1}
P	$\leftarrow$	$EXP\ [-Yb/(Xn + 1)]$ ---------> Exit P_{D1}

Figure 19: Algorithmic Program for Swerling 1 Target, P_{D1}

c. Swerling case 2 algorithm

Eq(63) can be modified as:

$$
\begin{aligned}
P_{D_2} &= 1 - I\left(\frac{Y_b}{\sqrt{N}((1+\bar{x})}, N-1\right) \\
&= P_{fa}\left(N, \frac{Y_b}{1+Nx}\right)
\end{aligned}
\tag{93}
$$

A computer independent algorithm to implement the above equation is shown in Figure 20.

```
Enter Yb ,N

Y     ←    Yb
X     ←    (1 + N * X)
X0    ←    Y/X
YM    ←    exp(-X0)
YMS   ←    YM
M     ←    0
M     ←    M + 1
N     :    M
YM    ←    X0*YM/M
YMS   ←    YMS + YM
PD2   ←    XMS   --------> Exit Pd2
```

Figure 20: Algorithmic Program for Swerling 2 Target, P_{D2}

d. Swerling case 3 algorithm

From Eq(75), the equation for probability of detection for case 3 is as follows:

$$P_{D3}=e^{-\frac{y}{1+\frac{Nx}{2}}}\frac{(1+\frac{(Nx)}{2}y)}{(1+\frac{Nx}{2})^{2}}\qquad for\ N=1$$

$$=\frac{Y_b^{N-2}e^{-Y_b}}{(N-2)!}\frac{2Y_b}{N\bar{x}+2}+P_{fa}(N-1,Y_b)+(\frac{N\bar{x}+2}{N\bar{x}})^{N-2}e^{\frac{-2Y_b}{N\bar{x}+2}} \qquad (94)$$

$$[1-\frac{2(N-2)}{N\bar{x}}+\frac{2Y_b}{N\bar{x}+2}][1-p_{fa}(N-1,\frac{Y_bN\bar{x}}{N\bar{x}+2})];\ N\geq 2$$

An algorithm for above equation is presented in computer independent notation in Figure 21.

```
Enter Yb,N,X

XN   <=  N*X
Y    <-  Yb
N    :   1
A    <-  0
YM   <-  EXP (-Y)
YMS  <-  YM
M    <-  0
M    <-  M + 1
N-1  <-  M
YM   <-  YM  Y/M
YMS  <-  YMS + YM
P2   <-  YMS
1    :   A
A    <-  1
P1   <-  P2
Y    <-  Y (1 +2/XN)
A1   <-  (1 + 2/XN)^(N-2)
A2   <-  EXP  (-2Yb/(XN+2))
A3   <-  1 - 2 (N-2)/(XN + 2Yb/(XN + 2)
P3   <-  A1 * A2 * A3 * (1 - P2)
B    :   1
2    :   N
L    <-  N - 2
C    <-  L
L    <-  L - 1
0    :   L
C    <-  (C*L)
B    <-  C
P4   <-  2Yb^(N-1) e^(-Yb)/B*(XN+2)
P    <-  P1 + P2 + P3  ----------> Exit  PD
G    <-  EXP  -2Yb/(XN+2)
H    <-  2 XN Yb/(XN+2)^2
P    <-  G (1 + H) ----------->Exit  PD
```

Figure 21: Algorithmic program for Swerling 3 Target, P_D

e. Swerling case 4 algorithm

Swerling case 4 computer program modified expression can be obtained by applying the power series in Eq (30) into Eq(81) as

$$P_{D_4}=1-\frac{N!}{(1+\overline{x}/2)^N}\sum_{k=0}^{N}\frac{\frac{\overline{x}}{2}^k[1-P(N+k,\frac{Y}{1+\overline{x}/2})]}{k!\,(N-k)!}$$

$$=1-(\frac{1}{1+\frac{Nx}{2}})^N[\sum_{m=N}^{2N-1}e^{-(\frac{Y}{1+\frac{Nx}{2}})}\frac{(\frac{Y}{1+\frac{Nx}{2}})^m}{m!}\sum_{K=0}^{m-N}\frac{N!\,(\frac{Nx}{2})^K}{K!\,(N-K)!} \tag{95}$$

$$+\sum_{m=2N}^{\infty}e^{-(\frac{Y}{1+\frac{Nx}{2}})}\frac{(\frac{Y}{1+\frac{Nx}{2}})^m}{m!}\sum_{K=0}^{N}\frac{N!\,(\frac{Nx}{2})^K}{K!\,(N-K)!}]$$

$$=1-(\frac{1}{1+\frac{Nx}{2}})^N[\sum_{m=N}^{2N-1}e^{-(\frac{Y}{1+\frac{Nx}{2}})}\frac{(\frac{Y}{1+\frac{Nx}{2}})^m}{m!}\sum_{K=0}^{N}\frac{N!\,(\frac{Nx}{2})^K}{K!\,(N-K)!}$$

$$+(1+\frac{Nx}{2})^N\sum_{m=2N}^{\infty}e^{-(\frac{Y}{1+\frac{Nx}{2}})}\frac{(\frac{Y}{1+\frac{Nx}{2}})^m}{m!}]$$

$$=\sum_{m=0}^{2N-1}e^{-(\frac{Y}{1+\frac{Nx}{2}})}\frac{(\frac{Y}{1+\frac{Nx}{2}})^m}{m!}$$

$$-\sum_{m=N}^{2N-1}e^{-(\frac{Y}{1+\frac{Nx}{2}})}\frac{(\frac{Y}{1+\frac{Nx}{2}})^m}{m!}[\sum_{K=0}^{m-N}\frac{N!}{K!\,(N-K)!}(\frac{\frac{Nx}{2}}{1+Nx/2})^K(\frac{1}{1+Nx/2})^{N-K}]$$

$$=P_{fa}(N,\frac{Y}{1+Nx/2})+\sum_{m=0}^{2N-1}e^{-(\frac{Y}{1+\frac{Nx}{2}})}\frac{(\frac{Y}{1+\frac{Nx}{2}})^m}{m!}$$

$$[1-\sum_{K=0}^{m-N}\frac{N!}{K!\,(N-K)!}(\frac{\frac{Nx}{2}}{1+Nx/2})^K(\frac{1}{1+Nx/2})^{N-K}]$$

with this expression the detection probability computation for this case therefore can be simplified according to equation (95) and avoid the computation of infinite power series. A computer independent algorithm is at Figure 22.

```
Enter Yb,N,X

     XN    ←   X
     M     ←   0
     MK    ←   (2/(XN +2))^N
     Y     ←   Yb
     B     ←   2Y/(XN + 2)
     YM    ←   e^-B
     YMS   ←   YM
     M     ←   M + 1
  ≤  N     :   M
     YM    ←   YM ∘ B/M
     YMS   ←   YMSS + YM
     SUM   ←   YMS
     MKS   ←   MK
     YM    ←   YM ∘ B/M
     SUM   ←   SUM + YM (1-MKS)
     MK    ←   XN ∘ MK ∘ (2N-M)/2(M-N+1)
     MKS   ←   MKS + MK
     M     ←   M + 1
  ≥  2N    :   M
     P4    ←   SUM    -------->Exit Pd4
```

Figure 22: Algorithmic program for Swerling 4 Target, P_{D4}

f. Swerling case 5 algorithm

In case 5, the expression of Marcum steady target model in Eq (38) can also be modified by substituting the infinite power series for the modified Bessel function $I_N(X)$ given by

$$I_N(x) = \left(\frac{x}{2}\right)^N \sum_{K=0}^{\infty} \frac{\left(\frac{x}{2}\right)^{2K}}{K!\,(N+K)\,!} \tag{96}$$

and interchanging the order of summation and integration, one obtain the probability detection for case 5 as:

$$P_{D5} = e^{-Nx} \sum_{K=0}^{\infty} \frac{(Nx)^K}{K!} \left[\sum_{m=0}^{N-1+K} e^{-Y} \frac{Y^m}{m!} \right] \tag{97}$$

By interchanging the order of the summation results in a efficient computation representation as:

$$P_{D5} = \sum_{m=0}^{N-1} e^{-Y_b} \frac{Y_b^m}{m!} + \sum_{m=N}^{\infty} e^{-Y_b} \frac{Y_b^m}{m!} \left(1 - \sum_{K=0}^{m-N} e^{-Nx} \frac{(Nx)^K}{K!}\right) \tag{98}$$

The right hand side of equation (98) has two terms, the latter term is a infinite summation of power series, therefore a recursive evaluation is necessary for computing P_{D5}.

The method to handle this infinite power series is to separate above equation into two terms. Let

$$P_{D5} = P_L + P_{\infty}; \tag{99}$$

where L represent the integrated pulses where L represent the integrated pulses. Let $L \geq N$, then P_L can be represented as

$$P_L=\sum_{m=0}^{N-1} e^{-Y_b}\frac{Y_b^m}{m!}+\sum_{m=N}^{L} e^{-Y_b}\frac{Y_b^m}{m!}\left(1-\sum_{K=0}^{m-N} e^{-Nx}\frac{(Nx)^K}{K!}\right) \tag{100}$$

the error term P_e can be represented as

$$P_e=\sum_{m=L+1}^{\infty} e^{-Y}\frac{Y^m}{m!}\left(1-\sum_{K=0}^{m-N} e^{-Nx}\frac{(Nx)^K}{K!}\right) \tag{101}$$

if let $P_L = P_{D5}$ then the truncated error term will be P_e

$$\begin{aligned} P_e &= \sum_{m=L+1}^{\infty} e^{-Y}\frac{Y^K}{m!}\left(1-\sum_{K=0}^{K-N}\frac{(Nx)^k e^{-Nx}}{k!}\right) \\ &\le \left(1-\sum_{m=0}^{L} e^{-Y}\frac{Y^K}{K!}\right)\left(1-\sum_{K=0}^{L+1-N}\frac{(Nx)^k e^{-Nx}}{k!}\right) \end{aligned} \tag{102}$$

Let L be large enough such that

$$\epsilon(L)\ge\left(1-\sum_{m=0}^{L} e^{-Y}\frac{Y^K}{K!}\right)\left(1-\sum_{K=0}^{L+1-N}\frac{(Nx)^k e^{-Nx}}{k!}\right) \tag{103}$$

Therefore

$$P_{D5}-P_L\le\epsilon \tag{104}$$

An upper bound can be found to limit the desired accuracy and avoid the unnecessary computation. The computer independent algorithm is in Figure 23.

```
enter Yb,N,S

XN    ←   X * N
YM    ←   EXP (-Yb)
YMS   ←   YM
M     ←   1
≦
N     :   M
YM    ←   YM ∘ Yb/M
YMS   ←   YMS + YM
M     ←   M + 1
SUM   ←   YMS
XB    ←   EXP (-XN)
XBS   ←   XB
YM    ←   YM ∘ Yb/M
YMS   ←   YMS + YM
SUM   ←   SUM + YM (1 - XBS)
M     ←   M + 1
K     ←   M - N
XB    ←   XB ∘ XN/K
XBS   ←   XBS + XB
≤ε    :   (1 - YMS)(1 - XBS)
P     ←   SUM  -------------->  eXIT WITH pDS
```

Figure 23: Algorithmic program for Swerling 5 Target, P_{DS}

2. Solving for round off error

Round off error can take place during the calculation of case 5. That is because during the calculation of probability of detection, if the signal-to-noise ratio x is zero, the output detection probability will become the probability of false alarm, then the criterion in Eq (103) will become

$$\begin{aligned}\epsilon(L) &\le \left(1-\sum_{m=0}^{L} e^{-Y}\frac{Y^{K}}{K!}\right)\left(1-\sum_{k=0}^{L+1-N}\frac{(Nx)^{k}e^{-Nx}}{k!}\right)\\ &\le (1-1)\left(1-\sum_{k=0}^{L+1-N}\frac{(Nx)^{k}e^{-Nx}}{k!}\right)\\ &\le 0\end{aligned} \tag{105}$$

Since we can not set the minimum acceptable error to be zero, the computation process will not terminate. Due to this reason, trade-off between the accuracy and the maximum value of output data should be made. An algorithm to compute the threshold y_b to deal with the round off error is in Figure 24. The corresponding MATLAB file name is **Marcum.m** which can be used to compute the maximum acceptable threshold level y_b when the value of signal-to-noise ratio is zero and the number of pulses is large.

```
<      Enter P_fa, Y_0
       Y      :    β
       M      ←    0
       YLX    ←    0
       YLOG   ←    LOG(Y)
       TSUM   ←    EXP(-Y)
       M      ←    M+1
       M      ←    N-1
       SUNK   :    LOG(M)
       YLX    ←    YLX+YLOG-SUNK
       TSUM   ←    TSUM+EXP(YLX-Y)
≥      Y_1    ←    Y+LOG(XP_FA*TSUM)*TSUM/EXP(YLX-Y)
       |Y-Y_1| :   ε
       Y_b    ←    Y_1---------------> Y_b
       Y      ←    Y_1
       M      ←    0
       YLX    ←    0
       YLOG   ←    LOG(Y)
<      Y-YLX  :    β
       M      ←    M+1
       SUNK   ←    LOG(N)
       YLX    ←    YLX+YLOG-SUNK
       TUM    ←    EXP(YLX-Y)
       Y_b    ←    Y_1 -----------------> Y_b
```

Figure 24: Algorithmic Program for Detection Thresholds, Y_b with round off error preventation

3. Solving for underflow and overflow

As mentioned before, the maximum acceptable number for MATLAB is nearly $e^{-709.782712893384040999999}$, the underflow and overflow problem arise due to the fact that the power series in M-S equations has the form

$$e^{-Y}\frac{Y^K}{m!} \quad ; \quad e^{-X}\frac{X^K}{m!} \tag{106}$$

If the calculation for this power series is over the maximum or below the minimum number that MATLAB can represent, then the MATLAB will return a string named NaN or empty matrix ([ ]) and the whole computation process will be meaningless. If the computation needed large integrated pulse number and large threshold or signal-to-noise ratio, a special effort has to be made to retain the accuracy. One method is let the series be of the form

$$e^{-y}\,\frac{y^k}{k!}=\exp\left[-y+k\ln(y)-\sum_{N=1}^{k}\ln(N)\right]=e^{-\beta} \tag{107}$$

and let the value $e^{-\beta}$ be compared to the smallest acceptable value $e^{-709.782712893384040999999}$ in MATLAB, of course $e^{-709.782712893384040999999}$ is much smaller than the tolerance number ϵ assumed in the program. If β is greater than 709.7821... then increase the number K in above equation until it is less than 709.7821... and then start the summation process. This method will save MATLAB computation of power series with a large number and avoid the under flow problems. An algorithm for using this method to compute the detection probability is in Figure (25), and the MATLAB source code is **threshm.m** and is given in Appendix B.

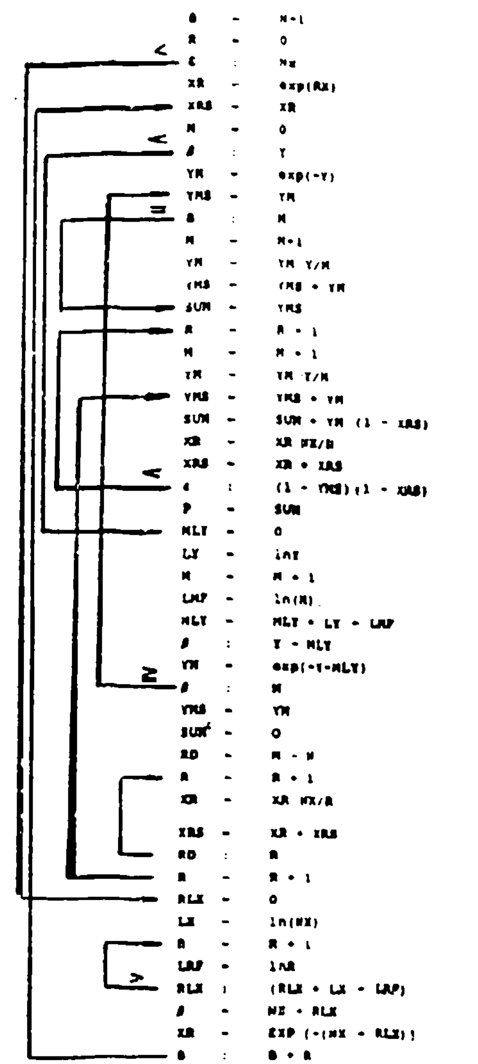

Figure 25: Algorithmic porgram for computing the probability of dtection, P_{d} with underflow preventation

4. Input arguments and output data

The table below summarizes all input and output data for the programs.

PROGRAM	INPUT DATA	OUTPUT DATA
THRESH.M	P_{FA} N	Y_b
PROB.M	Y_b ,N	P_{FA}
SWERL1.M	P_{fa},N, $\bar{x}$	P_{D1}
SWERL2.M	P_{fa},N, $\bar{x}$	P_{D2}
SWERL3.M	P_{fa},N, $\bar{x}$	P_{D3}
SWERL4.M	P_{fa},N, $\bar{x}$	P_{D4}
SWERL5.M	P_{fa},N, $\bar{x}$	P_{D5}
THRESHM.M.m	N, P_{fa}	Y_b
MARCUM.M	Y_b , N , $\bar{x}$	P_{D1} or P_{fa}
POINT.M	N, $\bar{x}$	Pd (numerical)
RADPOINT.M	N, $\bar{x}$	Pd (output data)
RADAR.M	P_{fa},N, $\bar{x}$	CHART S/N vs Pd
RADAR1.M	P_{fa},N, $\bar{x}$	CHART Pfa vs Pd
MEYER.M	P_{fa},N, $\bar{x}$	CHART N vs Pd
RANGE.1	P_{fa},N, $\bar{x}$,	HART R vs Pd

RANGE.3	p_{fa}, N, $\bar{x}$,	CHART R vs Pd

Table 1 : Input arguments and output data

Where

N=Number of pulses to be integrated

$\bar{X}$ =average signal-to -noise ratio

Pd= probability of detection

p_{fa}=probability of false alarm

R= the desired detection range

ns=Swerling case number

IV. RESULTS

The M-S detection probability curves can be obtain by applying the MATLAB programs in Chapter 3. The results can let the user evaluate the detection probability detection performance easily. The detection range curves can illustrate relationship between the detection probability and the detection range by input the properly detection loss.

A. PROBABILITY OF DETECTION CURVES

As mentioned previously, the M-S model arise from the different fluctuations of target cross section. MATLAB programs can be used to plot probability of detection versus per pulse signal-to-noise ratio for given probability of false alarm for five cases. It can also be asked to determine per pulse signal-to-noise ratio required for given P_D and P_{fa}. This required signal-to-noise ratio can be used to compute maximum detection range from the radar range equation.

Fig 26 and Fig 27 compare the five target models for a false alarm of 10^{-6}, and the number of integrated pulse as N=10 and N=100 respectively. When the probability is larger than 0.33, all four cases in which the target cross section is not constant requires greater signal-to-noise ratio than the constant cross section case. This increase in signal-to-noise-ratio will cause a reduction in detection range. Therefore if the characteristic of the target cross section are not properly take into account, the actual performance of the radar might not measure up to the performance which is predicted from the constant target cross section. A greater signal-to-noise ratio is required when the fluctuations are correlated pulse to pulse (case 1 and case 3) than when the fluctuations are uncorrelated pulse to pulse(case 2 and case 4). Fig 27 also indicates that when the number of integrated pulses is large, the case 2 and case 4 will approach to the constant target case (case 5).

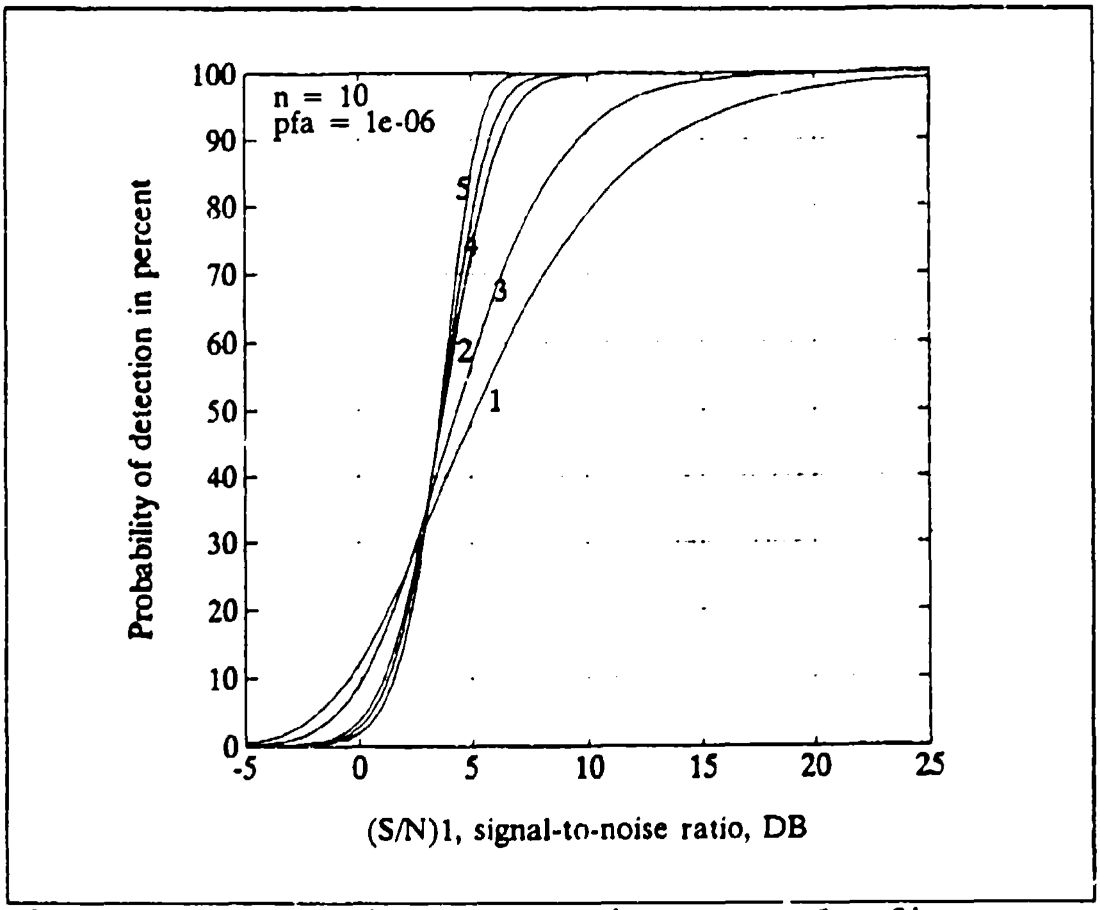

Figure 26: Probability of detection curves for five target models

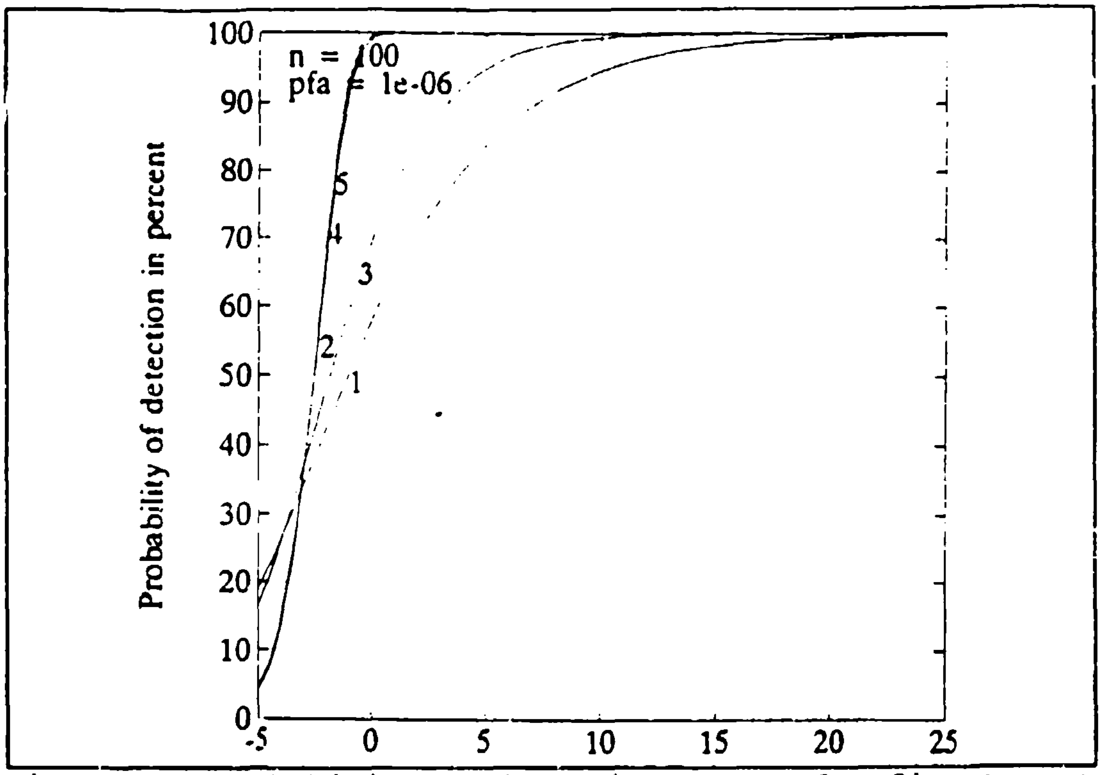

Figure 27: Probability of detection curves for five target models

Since both the false alarm rate P_{fa} and detection probability P_D are specified by the system requirement, the radar designer computes required signal-to-noise ratio from M-S curves and uses this to compute the maximum detection range. The greater the number of pulses integrated, the greater is resulting overall signal-to-noise ratio. This results in greater probability of detection, but it will require longer scan time. Figure 28 and Figure 29 give plots of P_d VS SNR for selected P_{fa}'s for case 1 for N equal to 100 and 500 respectively.

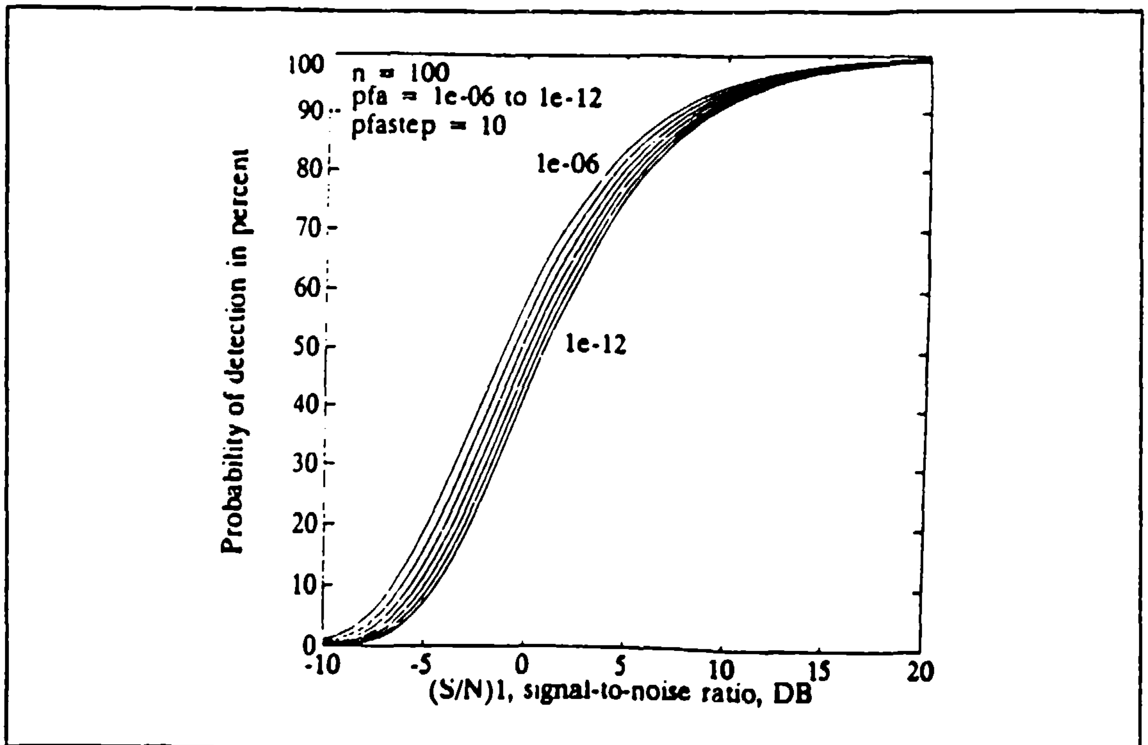

Figure 28: Probability of detection curves for Swerling case 1

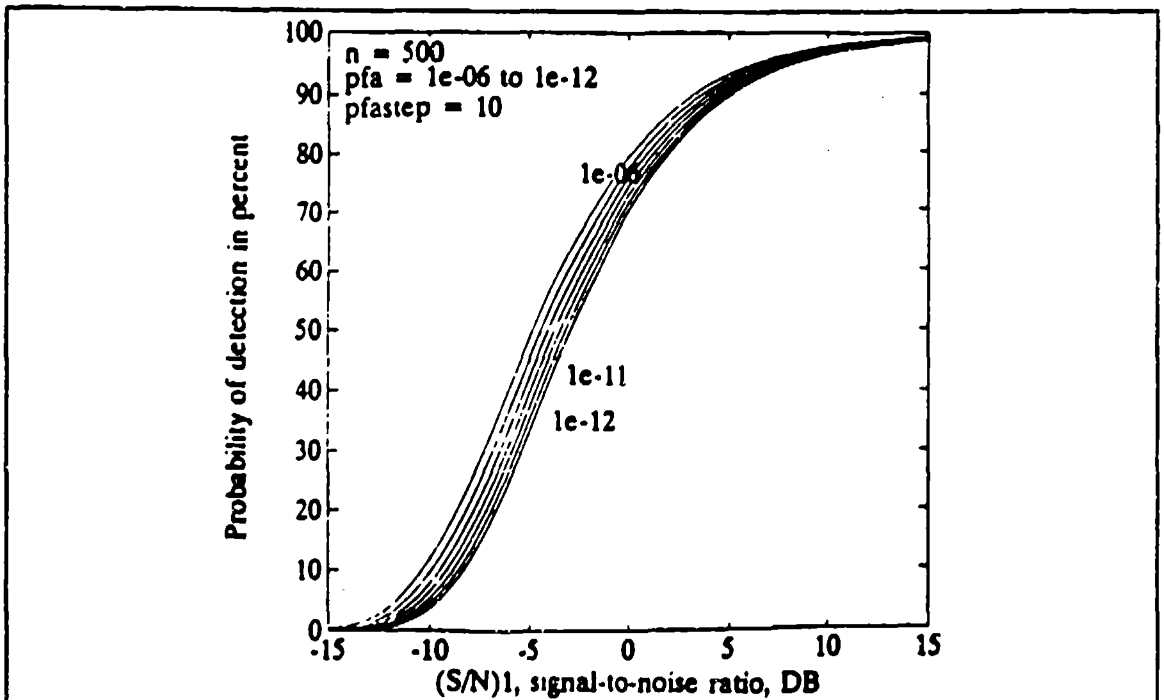

Figure 29: Probability of detection curves for Swerling case 1

Figure 30 and Figure 31 are the Swerling case 2 plots of P_d Vs SNR with 10 and 100 pulses integrated respectively. From Figure 30, signal-to-noise ratio of 4.8 db approximately is required to yield a probability of detection of 0.8 with 10 pulses integrated and probability of false alarm of 10^{-6}.

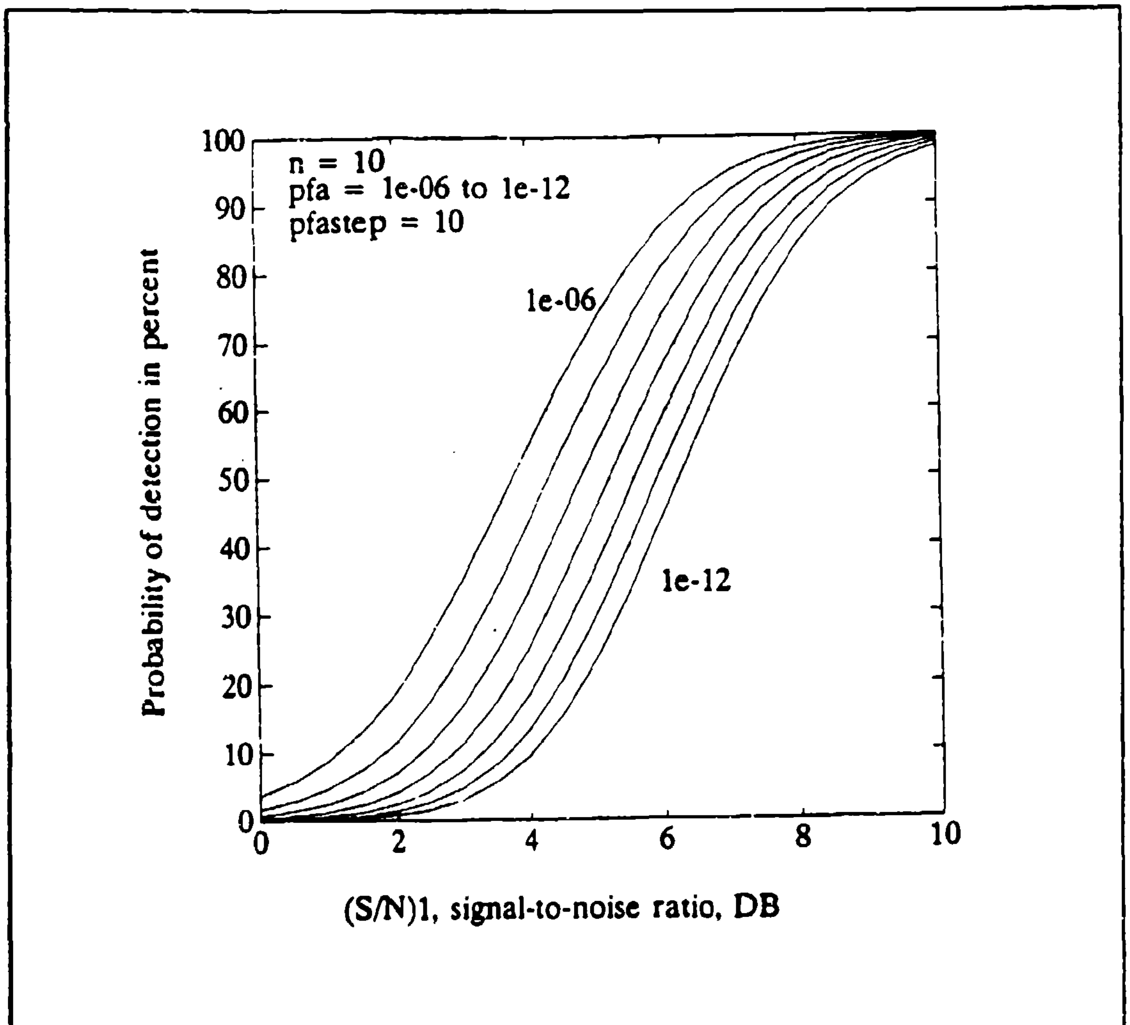

Figure 30: Probability of detection curves for Swerling case 2

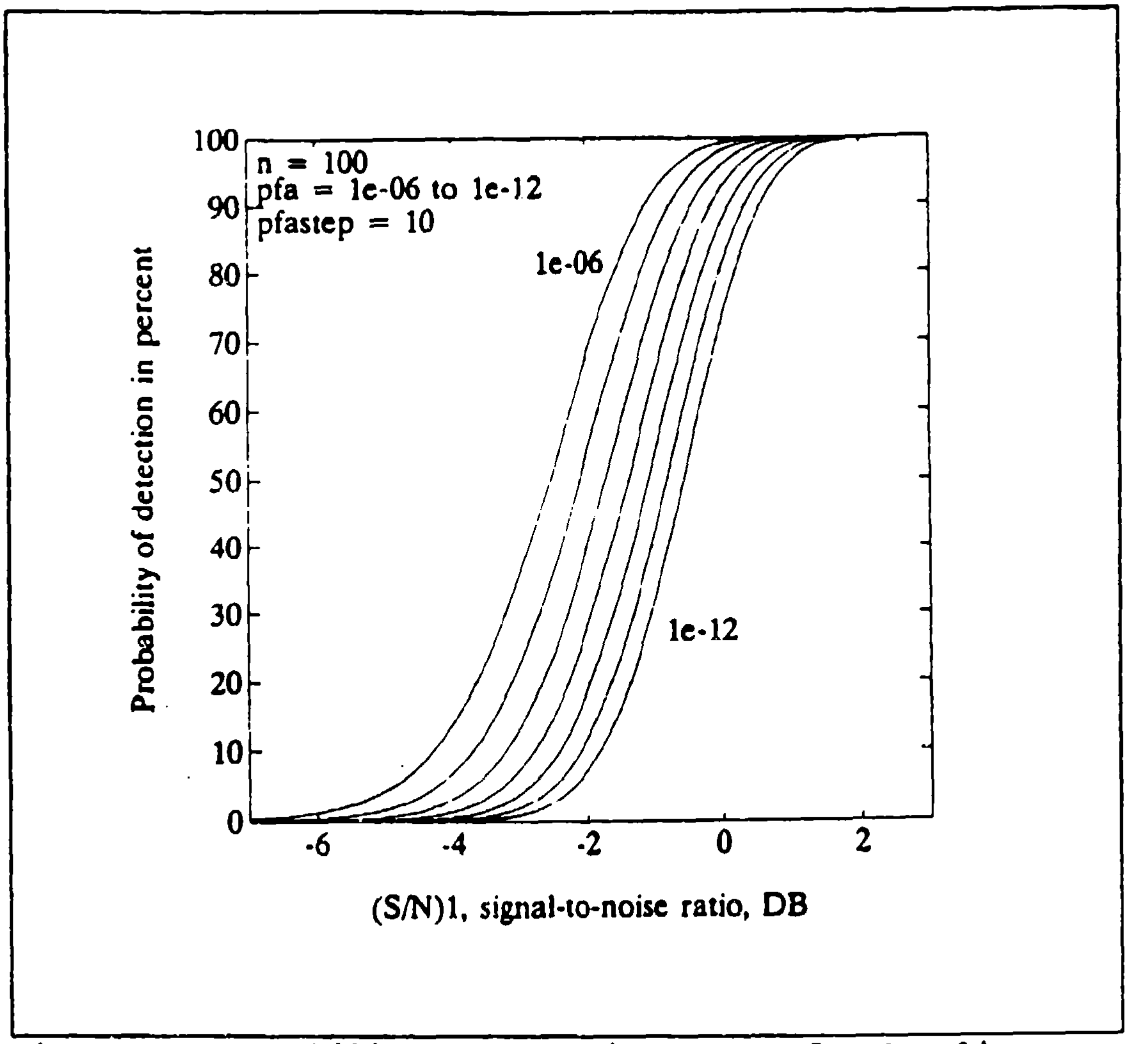

Figure 31: Probability of detection curves for Swerling case 2

Figure 32 and Figure 33 show the Swerling case 3 with 10 and 100 pulses integrated respectively.

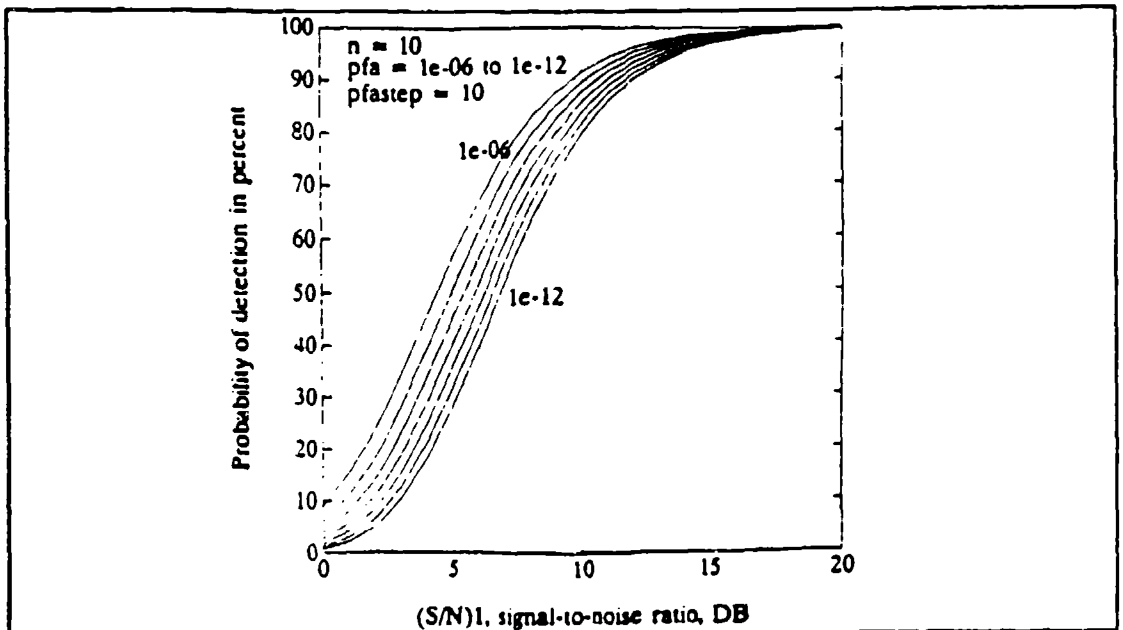

Figure 32: Probability of detection curves for Swerling case 3

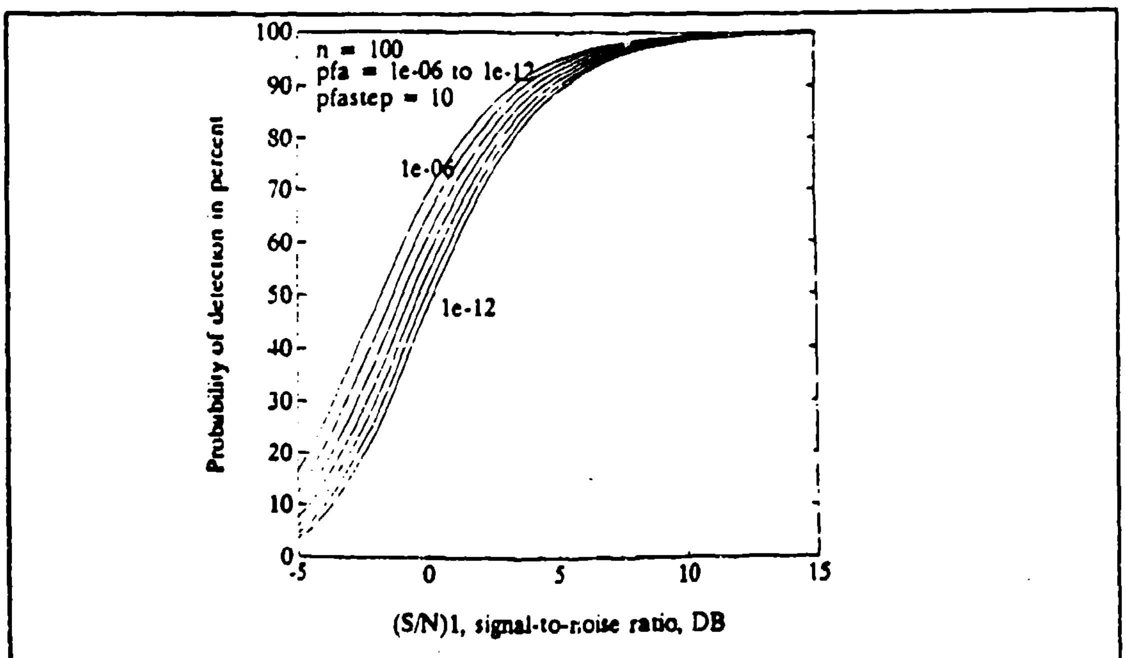

Figure 33: Probability of detection curves for Swerling case 3

Figure 34 and Figure 35 show Swerling case 4 pulse to pulse fluctuation with 10 and 100 pulses integrated respectively.

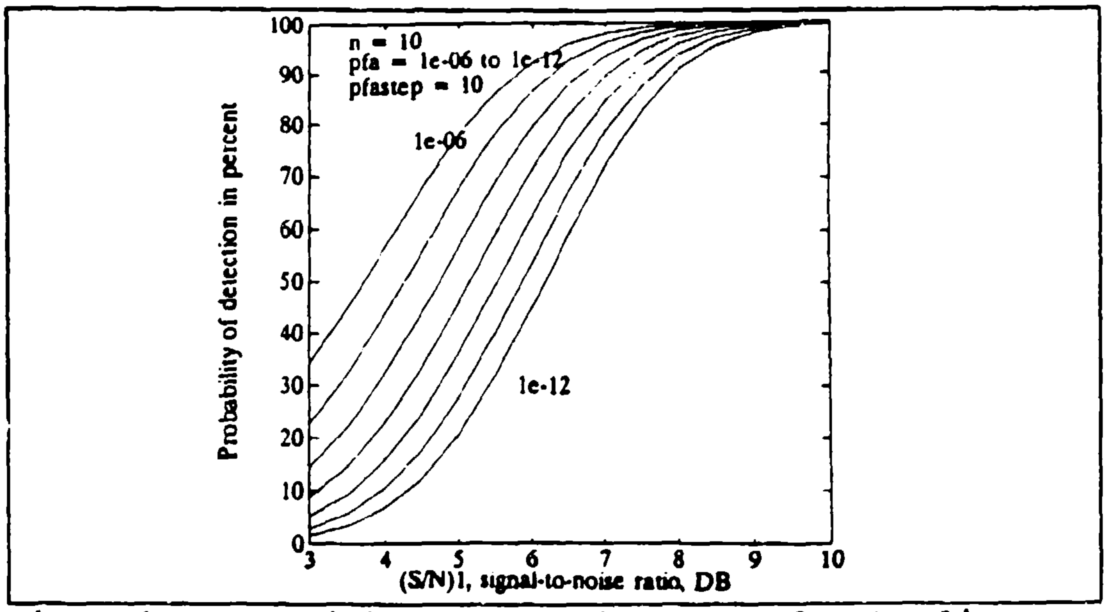

Figure 34: Probability of detection curves for Swerling case 4

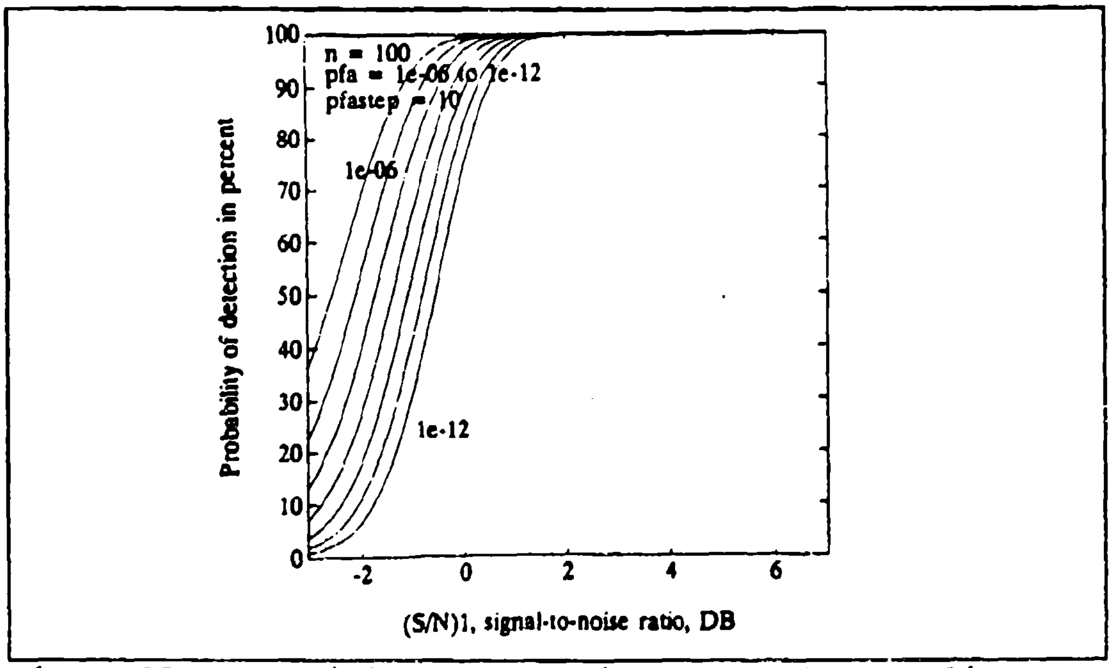

Figure 35: Probability of detection curves for Swerling case 4

Figure 36 and Figure 37 are Swerling case 5 steady-state target with 10 and 100 pulses integrated respectively.

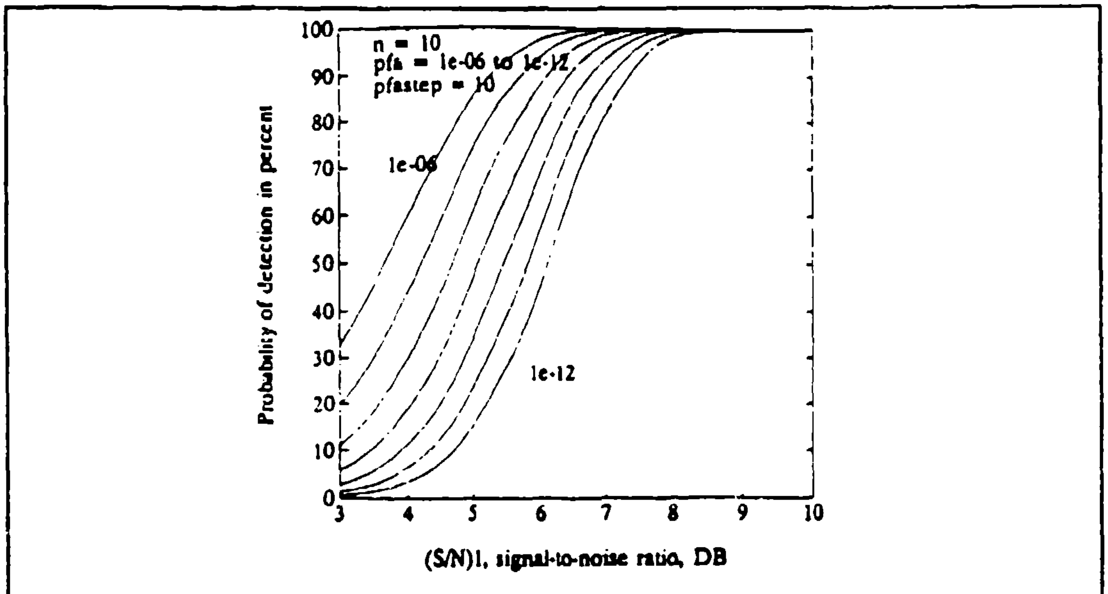

Figure 36: Probability of detection curves for Swerling case 4

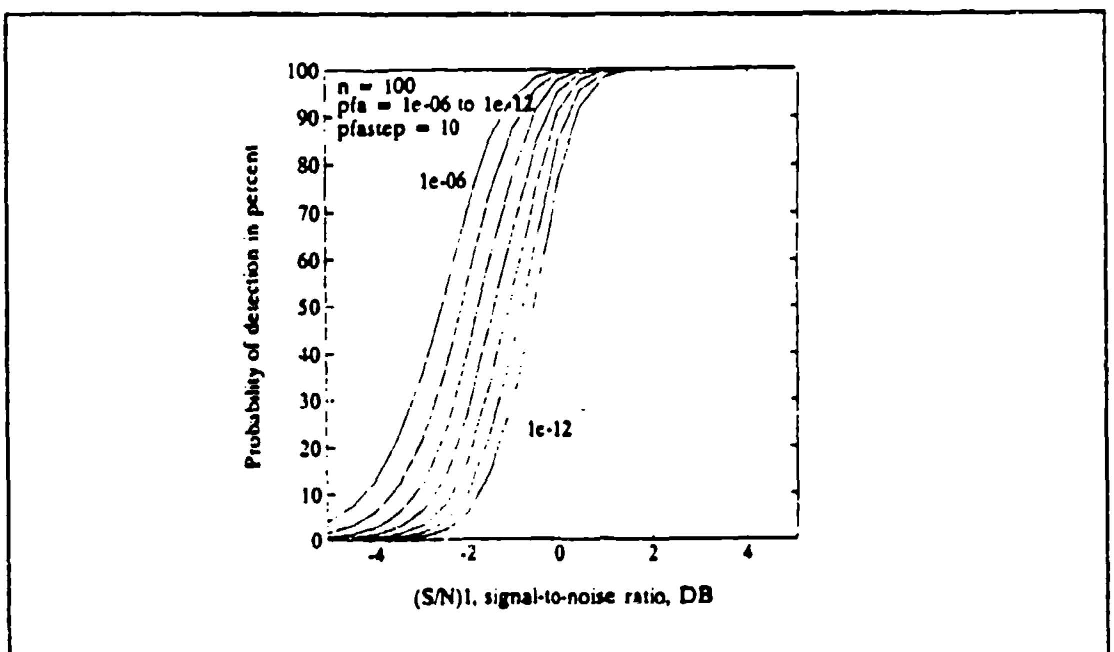

Figure 37: Probability of detection curves for Swerling case 4

B. THE DETECTIOON RANGE CURVES.

The ASR-9 radar is used as a sample to compute maximum detection range. ASR-9 radar is designed for surveillance of the terminal area around airports. A significant feature of this S-band radar is its use of moving target detection (MTD) processor which provides coherent integration among other features. The parameters of the ASR-9 radar are given in table below. Also tabulated is the empirically determined search detection range on a 1 m^2 target for the various Swerling models with a P_D =0.9 and P_{fa} =10^{-6}.

ASR-9 PARAMETERS AND RANGE CALCULATION

P_t=12oo kW	**ranges**	
f=2900 MHz	R_1 =39.41 nm	
τ=1.05 μs	R_2 =52.41 nm	
σ =1	R_3=46.68 nm	
G_t=33.5 db	R_4=57.29 nm	
G_r=33.5 dB	R_5=62.63 nm	
NF=5 dB		
L=12dB		
θ=1.3°	**Losses**	
$\dot{\theta}$=75°/sec		
PRF=1200 pps	Transmitter	2 dB
B_{Dopp}=150 Hz	Receiver	2 dB
P_D=0.9	Mismatch	1 dB
P_{fa}=10^{-6}	Integrator	1 dB
ρ=1	Collapsing	1 dB
	Beam Shape	3 dB
	MTI	2 dB
	Total	12 dB

Table 2 : ASR-9 PARAMETERS AND RANGE CALCULATION

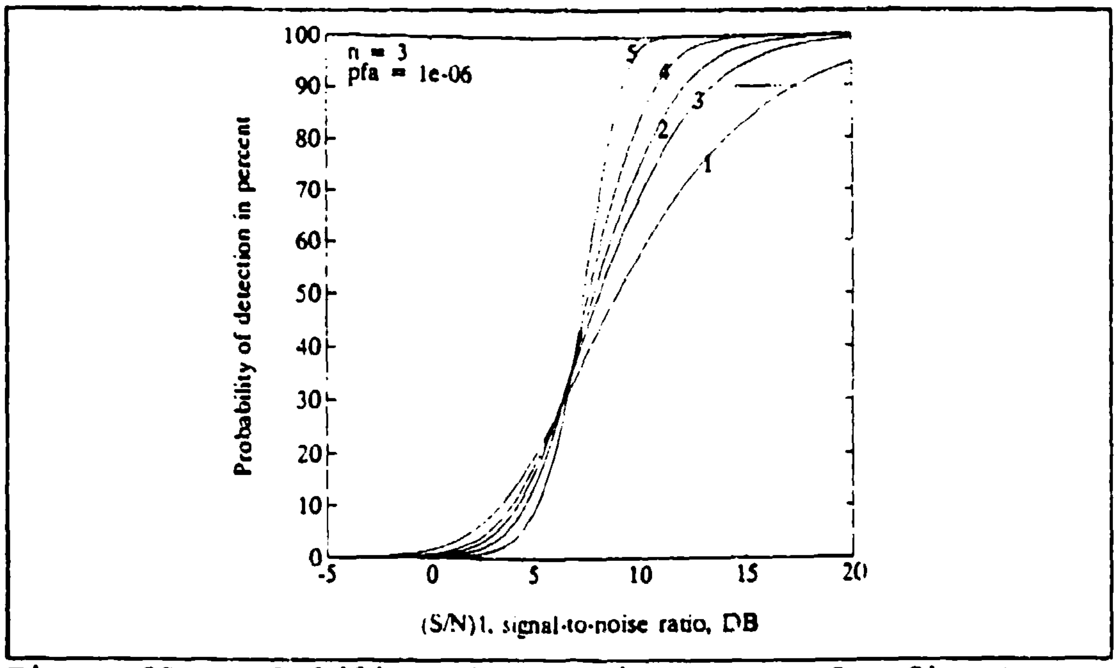

Figure 38: Probability of detection curves for five target models

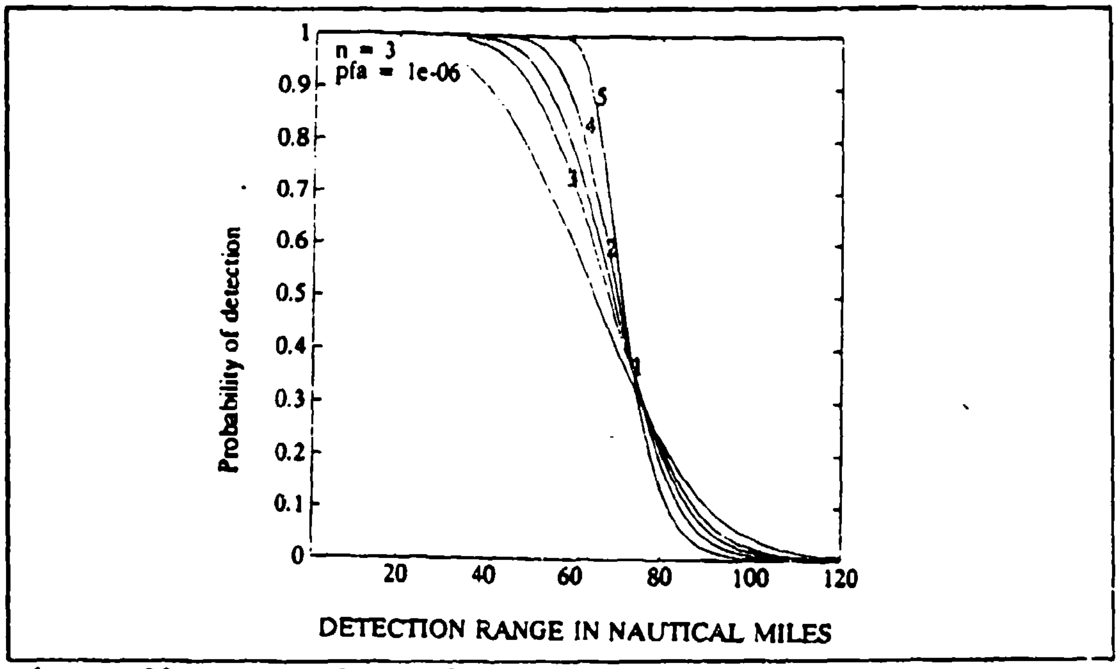

Figure 39: Range detection curves

C. **Numerical data output from probability of detection programs.**

A table of values below is included for the purpose of checking the programs.

N	Y_b	case	x=3.162278	x=10	x=31.62278	x=100	x=316.2278
1	13.8155055	1	.0361810926687	.2848035870497	.65475592794524	.8721557722127	.9573839597694
		2	.0361810926687	.2848035870497	.6547559279452	.8721557722127	.9573839539769 4
		3	.0202712698585	.2918843939447	.77944622707425	.9625961846032	.9959414970605
		4	.0202659173728	.2918820913596	.77946704886892	.9652566475676	.9959414102327
		5	0045852017166	.2480487019045	.99722505883065	.9999995192496	.9999995192469
3	19.1291681	1	.1971745188135	.5769062596383	.83647025963833	.9446918645347	.9821260632568
		2	.1630815180577	.7468903012970	.97820931643215	.9990169462157	.9999650689461
		3	.1784365463374	.5869964903258	.94510656525289	.9933271492948	.9992887535432
		4	.1368031960489	.8340491394154	.9977036152669	.9999940489495	.9999999917648
		5	.0888130400938	.9727252372625	.99999920934189	.9999992093418	.9999992093418

N	Y_b	case	x=3.1C2278	x=10	x=31.62278	x=100	x=316.2278
10	32.7013405	1	..4855434894507	.7911151202538	.9282416527967	.9765960615308	.9925329706042
		2	.73389703346785	.9989667533193	.99999988582414	.9999999999973	1
		3	.56937472704103	.9139452495156	.9800436879910	.9980084719731	.9998777462895
		4	.78178938715194	.9999629142286	.99999599999768	1	1
		5	.8355167894356	.9999991524271	.99999915242839	.9999991524283	.[illegible]9991524283

N	Y_b	case	x=3.162278	x=10	x=31.62278	x=100	x=316.2278
30	63.5481801	1	.69852634183006	.8917070566484	.96429073448000	.9885538115323	.99639654392453
		2	.99944729942719	.9999999999989	1	1	1
		3	.8147880711125	.9759019998584	.99728588213879	.99971801636709	.99997145717056
		4	.99993127276 3883	1	1	1	1
		5	.999998744645916	.9999992846098	.99999928460987	.99999928460987	.99999284460987

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

Radar detection theory is outlined in this thesis and computer programs are developed for probability of detection calculations in MATLAB. These programs are based on radar detection theory as developed by Marcum and Swerling. These programs give more accurate results than the commonly available programs based on the detectability method which is empirical is native. A user's guide and instruction are included in the thesis. Simple examples of the programs usage are illustrated in the thesis.

Students can use the programs written for this thesis to investigate radar system performance. These programs are cost effective, convenient to use and easy to reproduce since they are run on personal computers that are readily available to students. In particular, the program can aid students by removing a major computational burden to allow him to perform real world detection calculations. These programs can also be used for calculations in radar development work.

B. RECOMMENDATIONS

Due to an overflow underflow problem the programs are limited to integration of 600 pulses in the probability of detection calculations for all 5 cases. Most of the time this does not present any limitation as most radars integrate fewer than 600 pulses. However it may be desirable to remove this limitation by employing a Chernoff bound or other methods.

Appendix A

The gram-Charlier Series is a series expansion of a probability density function in terms of the Gaussian function, its derivatives and the moments of the original density function.

If $P(x)$ is a probability density function that is nearly Gaussian and x has zero mean and unit variance, then the Gram-Charlier series expansion of $P(x)$ is given by

$$P(x) = \sum_{i=0}^{\infty} a_i \, \phi^{(i)}(x) \quad \text{where} \tag{A-1}$$

$$\phi^{(0)}(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \tag{A-1.a}$$

$$\phi^{(i)}(x) = \frac{d^i \phi(x)}{dx^i}$$

Coefficient a_i can be evaluated in terms of the Hermite polynomials:

$$\phi^i(x) = \frac{(1)^i}{\sqrt{(2\pi)}} e^{(-x^2/2)} H_i(x) \tag{A-2}$$

Some typical polynomial are

$$\begin{array}{ll} H_0(x) = 1, & H_4(x) = x^4 - 6x^2 + 3 \\ H_1(x) = x, & H_5(x) = x^5 - 10x^3 + 15x \\ H_2(x) = x^2 - 1, & H_6(x) = x^6 - 15x^4 + 45x^2 - 15 \\ H_3(x) = x^3 - x, & H_7(x) = x^7 - 21x^5 + 105x^3 - 105x \end{array} \tag{A-3}$$

A recursive relation for these polynomials is

$$H_{i+1}(x) = xH_i(x) - iH_{i-1}(x) \qquad \textbf{(A-4)}$$

A particularly useful fact is that the hermite polynomials and the derivatives of the Gaussian function are biorthogonal. That is,

$$\int_{-\infty}^{\infty} \phi^i(x) H_j(x)\, dx$$

$$= (-1)^i \, i! \, \delta_{ij} \qquad \textbf{(A-5)}$$

$$= \begin{cases} (-1)^i \, i!, & i=j \\ 0 & i \neq j \end{cases}$$

To evaluate coefficient a_i, both side of Eq(A-1) are multiplied by $H_i(x)$ and result is integrated from $-\infty$ to ∞. With Eq(A-3), the result simplifies to

$$a_i = \frac{(-1)^i}{i!} \int_{-\infty}^{\infty} p(x) H_i(x)\, dx \qquad \textbf{(A-6)}$$

The first three coefficients a_0 a_1 and a_2 can be found by substituting Eq(A-3)into Eq(A-6)

$$a_0 = \int_{-\infty}^{\infty} P(x)\, dx = 1$$

$$a_1 = -\int_{-\infty}^{\infty} xp(x)\, dx = 0 \qquad \textbf{(A-7)}$$

$$a_2 = \frac{1}{2!} \int_{-\infty}^{\infty} (x^2-1) P(x)\, dx = \frac{1}{2}(1-1) = 0$$

With these results, the Gram-Charlier series expansion of $p(x)$ in Eq(A-1) can be simplified to

when the mean of random variable x is not zero and/or its

$$p(x) = \phi(x) + \sum_{i=3}^{\infty} \phi^{i}(x) \qquad \text{(A-8)}$$

variance σ is not unity, the Gram-Charlier expansion of $p(x)$ is obtained by expanding a related density function $g(t)$ in a Gram-Charlier series, where

$$t = \frac{x-\bar{x}}{\sigma} \qquad \text{(A-9)}$$

By a simple transformation of variables it follows that

$$\begin{aligned} p(x) &= \frac{g(t)}{\frac{dx}{dt}}\Big|_{t=(x-\bar{x})/\sigma} \\ &= \frac{1}{\sigma} g\left(\frac{x-\bar{x}}{\sigma}\right) \end{aligned} \qquad \text{(A-10)}$$

If the Gram-charlier series expansion for $g(t)$ is

$$g(t) = \sum_{i=0}^{\infty} C_i \phi^{(i)}(t) \qquad \text{(A-11)}$$

then from Eq(A-10) and Eq(A-11), the expression for $p(x)$ becomes

$$p(x) = \frac{1}{\sigma} \sum_{i=0}^{\infty} c_i \phi^{i}\left(\frac{x-\bar{x}}{\sigma}\right) \qquad \text{(A-12)}$$

If both sides of Eq(A-12) are multiplied by $H_j[(x-\bar{x})/\sigma]$ and the result integrated from $-\infty$ to ∞, coefficient c_i can be evaluated with relation (A-6) as

$$c_i = \frac{(-1)^i}{i!}\int_{-\infty}^{\infty} p(x)\, H_i\left(\frac{x-\overline{x}}{\sigma}\right) dx \qquad \textbf{(A-13)}$$

Evaluating Eq(A-13) for the first few coefficients c_i yields

$$\begin{aligned} &c_0 = 1, && c_4 = \frac{1}{4!}(\alpha_4 - 3) \\ &c_1 = c_2 = 0, && c_5 = -\frac{1}{(5!)}\alpha_5 - 10\alpha_3 \\ &c_3 = -\frac{1}{3!}\alpha, && c_6 = \frac{1}{(6!)}(\alpha_6 - 15\alpha_4 + 30) \end{aligned} \qquad \textbf{(A-14)}$$

where α_i is the ith central moment of $p(x)$, normalized by α^i:

$$\alpha_i = \frac{1}{\alpha_i}\int_{-\infty}^{\infty} (x-\overline{x})^i p(x)\, dx \qquad \textbf{(A-14)}$$

The α_i can be expressed in terms of the conventional moments of the distribution $p(x)$; thus if m_n denotes the nth moment of $p(x)$ that is

$$m_n = \int_{-\infty}^{\infty} x^n p(x)\, dx \qquad \textbf{(A-15)}$$

then α_3 through α_6 are related to noncentral moments m_n by

$$\begin{aligned} \alpha_3 &= \frac{m_3 - 3m_2m_1 + 2m_1^3}{\sigma^2} \\ \alpha_4 &= \frac{m_4 - 4m_3m_1 + 6m_2m_1^2 - 3m_1^4}{\sigma^4} \\ \alpha_5 &= \frac{m_5 - 5m_4m_1 + 10m_3m_1^2 - 10m_2m_1^2 + 4m_1^5}{\sigma^5} \\ \alpha_6 &= \frac{m_6 - 6m_5m_1 + 15m_4m_1^2 - 20m_3m_1^3 + 15m_2m_1^4 - 5m_1^6}{\sigma^6} \end{aligned} \qquad \textbf{(A-16)}$$

In summary, to obtain a Gram-Charlier series expansion, the moments m_n are found from Eq(A-16), or they can be conveniently computed from the characteristic equation of the distribution. Next the α_i are evaluated using Eq(A-17). The α_i are then substituted into Eq(A-14) to obtain coefficient c_i of Gram-Charlier expansion.

APPENDIX B

```
%FILE NAME: MS.M
clear
clg
clc
echo on
%THIS PROGRAM IS DESIGNED TO ALLOW THE STUDENT TO
%VARY THE PARAMETERS OF THE VARIOUS SWERLING CASES
%IN ORDER TO STUDY THE EFFECTS.
% CASE #:    DESCRIPTION
% 1. Returned pulses are of a constant amplitude over one
   scan, but are uncorrelated from  scan to scan.
% 2. Returned pulses are uncorrelated from pulse to pulse
   and correlated  from scan to scan.
% 3. Returned pulses are of a constant amplitude over one
   scan, but are uncorrelated from scan to scan.
% 4. Returned pulses are uncorrelated from pulse to pulse
   and correlated from scan to scan.
% 5. The static case with constant S/N and pulse amplitude
echo off
pause
clc
k=menu('MAIN MENU','THE M-S CURVES ',..
'NUMERICAL DETECTION PROBABILITY CALCULATION',..
'RANGE DETECTION CURVES')
if k==1
mscurve
elseif k==2
number
elseif k==3
rmenu
elseif k==4
density
end
%  Now restart the process with some different choices
%  Clear the workspace of unnecessary data to avoid
conflicts
clear i; clear sav; clear sdb; clear vv, clear pfa, clear
p;clear ns;
clear i; clear sav1; clear sdb1; clear vv, clear pfa1, clear
p;
%  Check and see if any info is to change
clc
echo on
%  If you want to restart the process       ENTER CHOICE = 1
%
%       or
```

```
%
%  To exit this programPRESS RETURN
echo off
choice = input('CHOICE = ')
if choice==1
   ms
else
end
clc
echo on
%   You have chosen to exit the program
%
%   Bye!
echo off



% FILE NAME: RADAR.M
clc
format long
echo
%*************************************************************
%     THIS PROGRAM RETURNS THE PLOTS FOR THE NUMBER
% OF PULSES AND SWERLING CASE SPECIFIED IN THE PARAMETERS.
% THE PLOTS WILL BE STORED IN METAFILES UNDER THE NAME
% "RADAR.MET" FOR AN EASY PRINT OUT.
%
%  (A)   the swerling case number has to be determined now
%*************************************************************
echo off
ns= input('Enter the case number you want to study ');

echo on
%*************************************************************
*
%  (B).   The number of radar pulses the program is to
%         integrate needs to be an integer between 1 and
600.
%*************************************************************
echo off
n=input('Number of Pulses to be integrated is n = ');
clc
echo on
%*************************************************************
%  (C).   The probability of false alarm rate curves (pfa)
to be plotted
%  must now be determined.  Each choice of a pfa will result
in a different
%  curve being plotted on the graph.  You need to choose the
following;
```

```
%    1.  The smallest pfa curve to be plotted, pfamin = ?
%
%    2.  The largest pfa curve to be plotted, pfamax = ?
%
%    3.  The step size between pfamin aand pfamax, pfastep =
?
%
%   If you wish to plot only one curve then enter the same
value for
%        pfamin and pfamax.
%
%        The suggested default step size to use is that of
PFASTEP = 10,
%        which is quite sufficient.  It is suggested that
pfamin and pfamax
%        be powers of 10 as that is the normal choice.
%**************************************************************
echo off

pfamin =input('pfamin = ')
pfamax =input('pfamax = ')
pfastep=input('pfastep = ')
clc
echo on
%**************************************************************
%  (D).   The signal to noise ratio (S/N) in dB for which
you wish to
%         plot needs to be determined.  The choices you need
%         to make are;
%
%    1.  The smallest S/N point to be plotted, sdbmin = ?
%
%    2.  The largest S/N point to be plotted, sdbmax = ?
%
%           Remember that S/N must be entered in dB.
%
%        3.      The stepsize between sdbmin and sdbmax = ?
%**************************************************************
echo off
sdbmin =input('In dB the sdbmin is  ')
sdbmax =input('In db the sdbmax is  ')
sdbstep= input('In db the sdbstep is ')
clg
clc
echo on
%-------------------------------------------------------------
% CHECK YOUR PARAMETERS !
--------------------------------------------------------------
echo off
fprintf('The case number you is %8 2f\n',ns);
fprintf('The number of pulses you choice are %8.2f\n',n);
```

```
fprintf('The max false alarm probability you choice is
%8.2e\n',pfamax);
fprintf('The min false alarm probability you choice
are%8.2e\n',pfamin);
fprintf('The pfa stepsize is %8.2f\n',pfastep');
fprintf('The max signal-to-noise ratio you choice is
%8.2f\n',sdbmax);
fprintf('The min signal-to-noise ratio %8.2f\n',sdbmin);
fprintf('The S/N stepsize is %8.2f\n',sdbstep');
echo on
%------------------------------------------------------------
%IF THE PARAMETERS ARE CORRECT PRESS 1
%IF THE PARAMETERS ARE NOT  CORRECT PRESS 2
%------------------------------------------------------------
echo off
choice=input('CHOICE= ');
if choice==2
radar
elseif choice==1
end
echo on
%************************************************************
%  Now run the calculations for the above data and plot it
%************************************************************
echo off
%delete radar.met
format long
nmax = 700;
axis('square');
vv=[sdbmin sdbmax 0  100];
axis(vv);
if n < 1 | n > 600 | n~=fix(n),
  error('Number of pulses input exceeds dimensions')
end
if ns < 1 | ns > 5 | ns~=fix(ns),
  error('Swerling case input does not correspond to
allowable choices')
end

pfa=pfamax;

sdb = sdbmin:sdbstep:sdbmax;
kk=length(sdb);
sav = 10 .^(sdb/10);
if ns == 1

%       swerling1
  while pfa >= pfamin
    for i= 1:kk
        %i = 1:sdbmax-sdbmin+1
        p(i)=swerl1(pfa,n,sav(i));
```

```
            p(i)=100*p(i);
         end

      plot(sdb,p,'-');grid;title('SWERLING 1');
      xlabel('(S/N)1, signal-to-noise ratio, DB'),
      ylabel('Probability of detection in percent ')
      text(sdbmin + 1/20*sdbmax,95 ,['n = ',int2str(n)])
      text(sdbmin + 1/20*sdbmax, 90,['pfa =
',num2str(pfamax),' to ',num2str(pfamin)])
      text(sdbmin + 1/20*sdbmax, 85,['pfastep =
',num2str(pfastep)]);
      gtext([num2str(pfa)]);
      hold on
      pfa = pfa/pfastep;
   end
   pause
   meta radar
elseif ns==2
%swerling2
   pfa=pfamax;
   while pfa >= pfamin
      for i= 1:kk
         % for i = 1:sdbmax-sdbmin+1
         p(i)=swerl2(pfa,n,sav(i));
         p(i)=100*p(i);
      end
      plot(sdb,p,'-'),title('SWERLING 2');
      grid, xlabel('(S/N)1, signal-to-noise ratio, DB'),
      ylabel('Probability of detection in percent')
      text(sdbmin + 1/20*sdbmax,95 ,['n = ',int2str(n)])
      text(sdbmin + 1/20*sdbmax,90,['pfa = ',num2str(pfamax),'
to ',num2str(pfamin)])
      text(sdbmin + 1/20*sdbmax,85,['pfastep =
',num2str(pfastep)])
      gtext([num2str(pfa)]);
      hold on
      pfa = pfa/pfastep;
   end
   pause;
   meta radar
elseif ns==3
%swerling3
   if n == 1
      error('algorithm for swerling 3 does not work for n=1')
   end
   while pfa >= pfamin
      for i=1:kk
         %for i = 1:sdbmax-sdbmin+1
         p(i)=swerl3(pfa,n,sav(i));
         p(i)=100*p(i);
      end
```

```
    plot(sdb,p,'-'),title('SWERLING 3');
    grid, xlabel('(S/N)1, signal-to-noise ratio, DB'),
    ylabel('Probability of detection in percent')
    text(sdbmin + 1/20*sdbmax,95 ,['n = ',int2str(n)])
    text(sdbmin + 1/20*sdbmax,90,['pfa = ',num2str(pfamax),'
to ',num2str(pfamin)])
    text(sdbmin + 1/20*sdbmax,85,['pfastep =
',num2str(pfastep)])
    gtext([num2str(pfa)]);
    hold on
    pfa = pfa/pfastep;
  end
  pause;
  meta radar
elseif ns==4
%swerling4
  while pfa >= pfamin
    for i=1:kk
      %for i = 1:sdbmax-sdbmin+1
      p(i)=swerl4(pfa,n,sav(i));
      p(i)=100*p(i);
    end
    plot(sdb,p,'-');title('SWERLING 4');
    grid; xlabel('(S/N)1, signal-to-noise ratio, DB');
    ylabel('Probability of detection in percent')
    text(sdbmin + 1/20*sdbmax,95 ,['n = ',int2str(n)])
    text(sdbmin + 1/20*sdbmax,90,['pfa = ',num2str(pfamax),'
to ',num2str(pfamin)])
    text(sdbmin + 1/20*sdbmax,85,['pfastep =
',num2str(pfastep)])
    gtext([num2str(pfa)]);
    hold on
    pfa = pfa/pfastep;
  end
  pause
  meta radar
else
%swerling5
  while pfa >= pfamin
    for i=1:kk
      % for i = 1:sdbmax-sdbmin+1
      p(i)=swerl5(pfa,n,sav(i));
      p(i)=100*p(i);
    end
    plot(sdb,p,'-'),title('SWERLING 5');
    grid, xlabel('(S/N)1, signal-to-noise ratio, DB'),
    ylabel('Probability of detection in percent')
    text(sdbmin + 1/20*sdbmax,95 ,['n = ',int2str(n)])
    text(sdbmin + 1/20*sdbmax,90,['pfa = ',num2str(pfamax),'
to ',num2str(pfamin)])
    text(sdbmin + 1/20*sdbmax,85,['pfastep =
```

```
',num2str(pfastep)])
    gtext([num2str(pfa)]);
    hold on
    pfa = pfa/pfastep;
  end
  pause
  meta radar
end
hold off


%FILE NAME: SWERL1.M
function pd=swerl1(pfa,n,sbar)
%*************************************************************
% RETURNS THE VALUE OF THE PROBABILITY OF DETECTION FOR THE
SWERLING1
% CASE, GIVEN THE NUMBER OF PULSES n, THE PROBABILITY OF
FALSE ALARM
% PFA AND THE AVERAGE SIGNAL TO NOISE RATIO S/N.
% EXAMPLE: SWERL1(PFA,N,SBAR);
%*************************************************************
format long
yb=thresh(pfa,n);
y=yb/(1+n*sbar);
cte=(1+1/n/sbar);
pd=prob(n-1,yb)+cte^(n-1)*exp(-y)*(1-prob(n-1,yb/cte));
return
end

% FILE NAME: SWERL2.M
%
*************************************************************
% RETURNS THE VALUE OF THE PROBABILITY OF DETECTION FOR THE
SWERLING2
% CASE, GIVEN THE NUMBER OF PULSES n, THE PROBABILITY OF
FALSE ALARM
% PFA AND THE AVERAGE SIGNAL TO NOISE RATIO S/N.
% EXAMPLE: SWERL2(PFA,N,SBAR);
%*************************************************************
function pd=swerl2(pfa,n,sbar)
yb=thresh(pfa,n);
y=yb/(1+sbar);
pd=prob(n,y);
return


% FILE NAME: SWERL3.M
%*************************************************************
% RETURNS THE VALUE OF THE PROBABILITY OF DETECTION FOR THE
SWERLING3
% CASE, GIVEN THE NUMBER OF PULSES n, THE PROBABILITY OF
```

```
FALSE ALARM
% PFA AND THE AVERAGE SIGNAL TO NOISE RATIO S/N.
% EXAMPLE: SWERL3(PFA,N,SBAR);
%*************************************************************
function pd=swerl3(pfa,n,sbar)
yb=thresh(pfa,n);
y=1;
for k=1:1:n-2;
y=yb/k*y;
end;
y1=2*yb/(n*sbar+2);
y=y*exp(-yb)*y1;
y2=((n*sbar+2)/n/sbar)^(n-2)*exp(-y1)*(1-2*(n-2)/n/sbar+y1);
y2=y2*(1-prob(n-1,yb*n*sbar/(n*sbar+2)));
pd=y+prob(n-1,yb)+y2;
return

% FILE NAME: SWERL4.M
%*************************************************************
% RETURNS THE VALUE OF THE PROBABILITY OF DETECTION FOR THE
SWERLING4
% CASE, GIVEN THE NUMBER OF PULSES n, THE PROBABILITY OF
FALSE ALARM
% PFA AND THE AVERAGE SIGNAL TO NOISE RATIO S/N.
% EXAMPLE: SWERL4(PFA,N,SBAR);
%*************************************************************
function pd=swerl4(pfa,n,sbar)
yb=thresh(pfa,n);
V=2*yb/(sbar+2);
ZK=(2/(sbar+2))^n;
YM=exp(-V);
YMS=YM;
for M=1:n-1
  YM=YM*V/M;
  YMS=YMS + YM;
end
SUM=YMS;
ZKS=ZK;
for M=n:(2*n-1)
  YM=YM*V/M;
  SUM = SUM + YM*(1-ZKS);
  ZK = sbar*ZK*(2*n-M)/(2*(M-n+1));
  ZKS = ZKS + ZK;
end
pd=SUM;
return;

%FILE NAME:SWERL5.M
%*************************************************************
% RETURNS THE VALUE OF THE PROBABILITY OF DETECTION FOR THE
SWERLING5
```

```
% CASE, GIVEN THE NUMBER OF PULSES n, THE PROBABILITY OF
FALSE ALARM
% PFA AND THE AVERAGE SIGNAL TO NOISE RATIO S/N.
% EXAMPLE: SWERL5(PFA,N,SBAR);
%**************************************************************
function pd=swerl5(pfa,n,sbar)
yb=thresh(pfa,n);
z=n*sbar;
YM=exp(-yb);
OK = 0;
k=n;
YMS=YM;
for i=1:k-1
  YM=YM*yb/i;
  YMS = YMS + YM;
end
SUM=YMS;
XB=exp(-z);
XBS=XB;
while OK == 0
  YM=YM*yb/k;
  YMS = YMS + YM;
  SUM= SUM+ YM*(1-XBS);
  e1=1-YMS;
  XB =XB*z/(k-n+1);
  XBS = XBS + XB;
  e2=1-XBS;
  er=e1*e2;
  if er <= 1e-6
    OK=1;
  end
  k = k + 1;
end
pd = SUM;
return;

% FILE NAME: THRESH.M
%**************************************************************
%* RETURNS THE THRESHOLD yb FOR A GIVEN PROBABILITY OF FALSE
   ALARM (pfa)
%* AND A GIVEN NUMBER OF PULSES
%* EXAMPLE: THRESH(Pfa,N).
%**************************************************************
function y=thresh(pfa,n)
format long
if n < 1 | n~=fix(n),
  error('Wrong input number of pulses');
  break
end
if n==1
  y=-log(pfa);
```

```
  return
end
l = -log10(pfa);
n2=sqrt(n);l2=sqrt(l);
y=n-n2+2.3*l2*(l2+n2-1);
comp=1e-6;
ratio=1;
while comp <= ratio
      p=prob(n,y);
      ym=1;
        for k=1:1:n-1;
          ym=ym/k*y;
        end;
      ym=ym*exp(-y);
      dely=(p/ym)*log(p/pfa);    % correction magitude
      y=y+dely;
      ratio=abs(dely/y);
end
return

% FILE NAME: PROB.M
%*****************************************************************
%*  THIS PROGRAM RETURNS THE FALSE ALARM PROBABILITY P(n,yb)
as in      *
%*  SUMMATORY(yb^k/fact(k)) for k=0 to n-1, AND IS A COMMON
FUNCTION     *
%*  FOR THE PROBABILITY OF DETECTION COMPUTATIONS
                *
%*  EXAMPLE: PROB(N,YB)
                *
%*****************************************************************
function p=prob(x,y)
if x< 0 | x~=fix(x),
  error('Number of pulses should be integer and greater than
zero'),break,end
inf==1.797693134862069*10^308;
E=709.782712893384040999999;
p=0;
if x==0
  return;
elseif x==1
  p = exp(-y);
  return;
else
%  [if x>1]
         t=1;
      for k=1:1:x-1;
          t=t/k*y;
          p=t+p;
      end
      p=(p+1)*exp(-y);
```

```
end
return;

%File name: THRESHM.M
function yb=threshm(pfa,n)
tol=1e-6;
E=709.78271289338404099999;
1 = -log10(pfa);
n2=sqrt(n);12=sqrt(1);
y=n-n2+2.3*12*(12+n2-1);

plog=log10(pfa);
xpfa= 1/pfa;
ok=0;
while ok==0

   if  y < E

         m=0;
         ylx=0;
         ylog=log(y);
         tsum=exp(-y);
              for m = 0:1:n-1
                     m=m+1;
                     sunk=log(m);
                     ylx= ylx+ylog -sunk;
                     tsum=tsum+exp(ylx-y);
               end
          yl=y+log(xpfa*tsum)*tsum/exp(ylx-y);

              if abs(y-yl) < tol
                    yb=yl;
                    ok=1;
                    return
              else
                    y=yl;
                    ok=0;
              end


    else
           m=0;
           ylx=0;
           ylog=log(y);
           okk=0;
           while okk==0

                  m=m+1;
                  sunk=log(m);
                  ylx=ylx+ylog-sunk;
```

```
            if y-y1x <= E

               tsum=exp(y1x-y);
               okk=1;
            else
               okk=0;
            end
         end
         if m >= n-1
            y1=y+log(xpfa*tsum)* tsum/exp(-y+y1x);
                 if abs(y1-y) < tol
                    yb=y1;
                    ok=1;
                    break;
                    return
                 else
                    y=y1;
                    ok=0;
                 end

         else
                for m=m:1:n-1
                    sunk=log(m);
                    y1x=y1x+ylog-sunk;
                    tsum=tsum+exp(-y+y1x);
                end
                y1=y+log(xpfa*tsum)*tsum/exp(-y+y1x);
                if  abs(y1-y) <= tol
                    yb=y1;
                    ok=1;
                    break;
                    return
                else
                    y=y1;
                    ok=0;
                end
         end
  end
end

%FILE NAME: MARCUM.M
function [p,ymo]=bound(n,y,x)
format long
err=1e-6;
E=709.78271289338404099999;

%******    start the program ****************************

lamda1=(n/(2*y))^2+(n*x)/y;
lamda=1-(n/(2*y))-sqrt(lamda1);
aux1=(-lamda*y)+(n*x*lamda)/(1-lamda);
```

```
aux2=exp(aux1)/(1-lamda)^n;
if aux2 <= err
   aux3=n*(x+1);
     if y >= aux3
        p=0;return
     else
        p=1;return
     end
else

ib=n-1;
k=0;
x=n*x;
if x <= E
   xk=exp(-x);
   xks=xk;
   m=0;
       if y<= E
          ym=exp(-y);
          yms=ym;
%-----------------------------------------------

            for m=1:n-1
                ym=ym*y/m;
                yms=yms+ym;
                    if err>(1-yms)
                       p=yms;
                       ymo=ym
                       x=x/n;

                       return
                     end
              end
          m=n-1;
          sum=yms;
          ymo=ym
          ok=0;
          while ok==0
                k=k+1;
                m=m+1;
                ym=ym*y/m;
                yms=yms+ym;
                sum=sum+ym*(1-xks);
                xk=xk*x/k;
                xks=xks+xk;
                ans=(1-yms)*(1-xks);
                    if  ans<= err
                        ok=1;
                     end
           end
```

```
        p=sum;ymo=ym;
    else
       ymly=0;
       ylog=log(y);
       okk=0;
       while okk==0
             m=m+1;
             xmf=log(m);
             ymly=ymly+ylog-xmf;
                   if (y-ymly) <= E
                      ym=exp(-y+ymly);
                      okk=1;
                    else
                       okk=0;
                    end
         end
         if  m <= ib
           yms=ym;
%-----------------------------------------------------------
              for  ml=m:n-1
                   ym=ym*y/m;
                   yms=yms+ym;
                   if err>(1-yms)
                      p=yms;
                      ymo=ym
                      x=x/n;
                      ook=1;
                      return
                    end

             end
           sum=yms;
           ymo=ym
           ok=0;
           while ok==0;
              k=k+1;
              m=m+1;
              ym=ym*y/m;
              yms=yms+ym;
              sum=sum+ym*(1-xrs);
              xk=xk*x/k;
              xrs=xrs+xk;
              ans=(1-yms)*(1-xrs);
                   if ans > err
                      ok=0;
                   else
                      p=sum;
                      x=x/n;
                      break;
                      return;
```

```
                    ok=1;
                end
            end
            p=sum;ymo=ym;

          else      %------------  for m> ib;
              ymo=0;return
              yms=ym;
              sum=0;
              kd=m-n;
              if  kd > x
                  p=0;
                  x=x/n;
                  break;
                  return;
               else
                  okkk=0;
                  while okkk==0;
                        k=k+1;
                        kk=xk*x/k;
                        xks=xks+xk;
                          while (k-kd) < 0
                                okkk=0;
                           end
                   end
                   ooook=0;
                   while ooook==0;
                         sum=sum+ym*(1-xrs);
                         xk=xk*x/k;
                         xrs=xrs+xk;
                         ans=(1-yms)*(1-xrs);
                         while ans < err
                               p=sum;
                               x=x/n;
                               ooook=1;
                               break;
                               return
                         end
                         k=k+1;
                         m=m+1;
                         ym=ym*y/m;
                         yms=yms+ym;
                         ooook=0;
                   end

               end
          end
     end

else
   xklx=0;
```

```
   xlog=log(x);
   ooook=0;
   while oook==0
      k=k+1;
      xklx==xklx+xlog-xkf;
      while (x-xklx) > E
               oook=0;
      end
   end
   xk=exp(-x+xklx);
   ib=ib+k;
%------------------------------
   xks=xk;
   m=0;
         if y<= E
             ym=exp(-y);
             yms=ym;
%----------------------------------------------
                for m=1:n-1
                     ym=ym*y/m;
                     yms=yms+ym;
                          if err>(1-yms)
                              p=yms;
                              ymo=ym;
                              x=x/n;
                              return
                           end
                 end
             sum=yms;
             ymo=ym;
             ok=0;
             while ok==0;
                     k=k+1;
                     m=m+1;
                     ym=ym*y/m;
                     yms=yms+ym;
                     sum=sum+ym*(1-xrs);
                     xk=xk*x/k;
                     xrs=xrs+xk;
                     ans=(1-yms)*(1-xrs);
                          if ans<= err
                               ok=1;
                          end
              end
              p=sum;

         else
             ymly=0;
             ylog=log(y);
             okk=0;
             while okk==0
```

```
        m=m+1;
        xmf=log(m);
        ymly=ymly+ylog-xmf;
              if (y-ymly) <= E
                  ym=exp(-y+ymly);
                  okk=1;
               else
                  okk=0;
               end
   end
   if  m <= ib
     yms=ym;

%-----------------------------------------------
       ook=0;
       while ook==0
         while m~=ib
             m=m+1;
             ym=ym*y/m;
             yms=yms+ym;
             if err>(1-yms)
                p=yms;
                ymo=ym;
                x=x/n;
                ook=1;
                break;
                return
               else
                ook=0;
               end
           end
         end
     sum=yms;
     ymo=ym;
     ok=0;
     while ok==0;
        k=k+1;
        m=m+1;
        ym=ym*y/m;
        yms=yms+ym;
        sum=sum+ym*(1-xrs);
        xk=xk*x/k;
        xrs=xrs+xk;
        ans=(1-yms)*(1-xrs);
            if ans > err
                ok=0;
            else
                p=sum;
                x=x/n;
                break;
                return;
```

```
                ok=1;
            end
        end


    else      %------------  for m> ib;
        ymo=0;
        yms=ym;
        sum=0;
        kd=m-n;
        if  kd>x
            p=0;
            x=x/n;
            break;
            return;
         else
            okkk=0;
            while okkk==0;
                    k=k+1;
                    kk=xk*x/k;
                    xks=xks+xk;
                      while (k-kd) < 0
                              okkk=0;
                       end
              end
              ooook=0;
              while ooook==0;
                      sum=sum+ym*(1-xrs);
                      xk=xk*x/k;
                      xrs=xrs+xk;
                      ans=(1-yms)*(1-xrs);
                      while ans < err
                              p=sum;
                              x=x/n;
                              ooook=1;
                              break;
                              return
                       end
                       k=k+1;
                       m=m+1;
                       ym=ym*y/m;
                       yms=yms+ym;
                       ooook=0;
               end

           end
       end
   end

end
end
```

LIST OF REFERENCES

1. J. Marcum, "A Statistical Theory of Target Detection by Pulsed Radar," *Institute of Radio Engineer Transations n information theory.*, Vol. IT-6, No. 2, April 1960

2. J. Marcum, "A Statistical Theory of Target Detection by Pulsed Radar Mathematical Appendix," *Institute of Radio Engineer Transations on information theory.*, Vol. IT-6, No. 2, April 1960

3. P. Swerling, "Probability of Detection for Fluctuating Targets," *Institute of Radio Engineer Transations on information theory.*, Vol. IT-6, No. 2, April 1960

4. D. Meyer and H. Meyer, *Radar Target Detection Handbook of Theory and Practice*, Academic Press, New York, 1973.

5. G. A. Campbell and R. M. Foster, *Fourier integrals for Practical Applications.*, Van Nostr and Reinhold, Princeton, New Jersey, 1947 ; Bell Telephone Syst. Monograph No. B-584

6. Lamont V. Blake, *Radar Range Performance Analysis,* D. C Heath and company Lexington, MA, 1980

7. Anthony D. Whalen, *Detection of Signal in Noise*, Academic Press, Inc. New York, 1971

8. M.I Skolnik, *Introduction to Radar System*, McGrow-Hill,New York, 1980

9. J. DiFranco and W. Rubin, *Radar Detection*, Arthech House, Dedham, MA, 1968

10. D. Curtis Schleher, *Introduction to Electronic Warfare,* Artech House Inc, 1986

11. The Math Works Inc, *The Student edition of MATLAB,* Prentice-Hall, Inc. A Simen & Schuster Company Englewood Cliffs, New Jersey, 1992.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria VA 22304-6145	2
2.	Library, Code 052 Naval Postgraduate School Monterey CA 93943-5002	2
3.	Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgurate School Monterey, CA 93943-5121	1
4.	Professor G.S. Gill, Code EC/Gl Department of Electrical and Computer Engineering Naval Postgurate School Monterey, CA 93943-5121	2
5.	Professor R. Janaswamy, Code EC/Js Department of Electrical and Computer Engineering Naval Postgurate School Monterey, CA 93943-5121	1
6.	Naval Academy Library Kao-Hsiung Tso-Ying P.O BOX 90175 Taiwan, R.O.C.	2
7.	Chung Cheng Institute of Technology Taoyuan, Taiwan, R.O.C.	1
8.	War Game Armed Forces University Republic of China P.O BOX 90040-16 Daje Taipei Taiwan	2